Departament de Teoria
del Senyal i Comunicacions
UNIVERSITAT POLITÈCNICA DE CATALUNYA

SMOS End-to-end Performance Simulator 4.0
User's Manual

REF: SO-UM-UPC-PLM-0006 02

Written by: M. Zapata (UPC)
A. Camps (UPC)
I. Corbella (UPC)
V. Barrena (GMV)
M. Zundo (ESA/ESTEC)

Revised by: A. Camps (UPC)
C. García (EADS-CASA)

Approved by: A. Camps (UPC)

January 23rd, 2006

INDEX

LIST OF ACRONYMS 5

PREFACE 7

CHAPTER I. ESA'S SMOS EARTH EXPLORER OPPORTUNITY MISSION 9
- I.1. The SMOS/MIRAS Mission 9
- I.2. Basic Concepts on Radiometry 9
- I.3. Basic Concepts on Aperture Synthesis Radiometry 12
- I.4. Imaging Considerations in Aperture Synthesis Radiometry 15

INSTALLATION 19

CHAPTER II. COMPUTER REQUIREMENTS 23
- II.1. Operating Systems 23
- II.2. MATLAB and toolboxes versions 24
- II.3. Hardware recommendations 25
- II.4. Installation description 26
  - II.4.1. Recompiling MEX functions 26
  - II.4.2. Libxml2 library installation 27
  - II.4.3. Running the SEPS v4.0 in MacOS 29
- II.5. Other questions 30

CHAPTER III. START-UP OF MATLAB & SEPS v4.0 33
- III.1. MATLAB Environment 33
- III.2. Updating the Search Path 34
- III.3. Installing and Configuring SEPS v4.0 38
- III.4. A walk through a basic SEPS v4.0 session 41

SEPS DESCRIPTION 43

CHAPTER IV. SCENARIO MANAGEMENT 47
- IV.1. New Scenario 48
- IV.2. Load Scenario 49
- IV.3. Save Scenario 50
- IV.4. Delete Scenario 51
- IV.5. Rename Scenario 51
- IV.6. Automatic Scenario 52
- IV.7. Batch Mode 60
- IV.8. Help 64

CHAPTER V. SCENARIO DEFINITION 65
- V.1. Satellite Definition 65
  - V.1.1. Orbit 66
  - V.1.2. Attitude 72
- V.2. Instrument Definition 75
  - V.2.1. Antenna Characterisation 75
  - V.2.2. Receiver and Calibration 85

V.2.3. Failure of Components 91
V.3. Image Reconstruction 96
V.3.1. UPC Algorithms 96
V.3.2. Other Algorithms 98
V.4. Simulation Parameters 99

CHAPTER VI. MISSION SIMULATION 110
VI.1. Mission Simulation Interface 110
VI.2. Monitoring a simulation 115

CHAPTER VII. FOM SIMULATION POSTPROCESSING 121
VII.1. Platform Figures Of Merit 122
VII.1.1. Osculating Elements Evolution 123
VII.1.2. Mean Elements Evolution 125
VII.1.3. Platform Attitude Evolution 126
VII.1.4. Subsatellite Point 127
VII.1.5. LTAN and Relative Position 128
VII.2. Instrument 129
VII.2.1. Antenna Patterns 130
VII.2.2. Antenna Patterns Differences 134
VII.2.3. Receiver Response 139
VII.2.4. System Temperatures 144
VII.2.5. Visibilities 146
VII.3. Brightness Temperature Maps 149
VII.3.1. Director Cosines Domain 152
VII.3.2. Cartography 161
VII.3.2.1. Functionality 162
VII.3.2.1.1. Cartographic Representation Method window 162
VII.3.2.1.2. Main Menu 165
VII.3.2.1.3. Show Snapshot # 165
VII.3.2.1.4. File 167
VII.3.2.1.5. Cursor Position 169
VII.3.2.1.6. Axes On/Off 171
VII.3.2.1.7. Change Scale 173
VII.3.2.1.8. Change Map Origin 176
VII.3.2.1.9. Zoom 177
VII.3.2.2. Projections 178
VII.3.2.2.1. Gnomonic Projections 179
VII.3.2.2.2. Stereographic Projections 180
VII.3.2.2.3. Orthographic Projections 181
VII.3.2.2.4. Cylindrical Developments 183
VII.3.2.2.5. Universal Transverse Mercator Projection 187
VII.3.2.2.6. Conical Developments 190
VII.3.2.3. Types of Data 191
VII.3.2.3.1. Initial types of data 192
VII.3.2.3.2. Projecting other types of data 193
VII.3.2.3.3. Data evolution along the simulation for a particular pixel 197

CHAPTER VIII. FIELD OF VIEW CALCULATOR 200
VIII.1. Radiometric Accuracy 201
VIII.1.1. Input Parameters 201
VIII.1.2. Outputs 208
VIII.2. Radiometric Sensitivity 210
VIII.2.1. Input Parameters 210
VIII.2.2. Outputs 211
VIII.3. Angular Resolution 212
VIII.3.1. Input Parameters 212
VIII.3.2. Outputs 214
VIII.4. FOV Properties 219
VIII.4.1. Input Parameters 220
VIII.4.2. Outputs 221
VIII.5. Reliability 226
VIII.5.1. Input Parameters 226
VIII.5.2. Outputs 228

CHAPTER IX. APPLICATIONS 230
IX.1. Output ASCII and binary data files 230
IX.2. Science and ancillary packets 237
IX.3. Changing system's definition files 239

APPENDICES: ADVANCED TOPICS 242

APPENDIX A. MODIFYING RECEIVERS' PARAMETERS 244
A.1. Antenna 246
A.2. Subsystems defined by S-parameters 248
A.3. Subsystems defined by voltage gain (and input impedance) 249
A.4. Subsystems with specific noise descriptions (amplifiers) 250
A.5. Frequency converter (I/Q mixer) 250
A.6. RF filter 251
A.7. IF filter 252
A.8. NIR 252
A.9. PMS 253

APPENDIX B. MODIFYING THE BRIGHTNESS TEMPERATURE GENERADOR 254
B.1. Bare soil 255
B.2. Vegetation covered soils 255
B.3. Snow covered soils 256
B.4. Frosted snow soil 258
B.5. Water bodies (oceans, seas and lakes) 259
B.6. Iced oceans 260
B.7. Atmosphere 261
B.8. Ionosphere 262

APPENDIX C. USING SEPS TO ESTIMATE THE RMS RADIOMETRIC ACCURACY (PIXEL BIAS) SENSIBILITY TO INSTRUMENTAL ERRORS 266

APPENDIX D. FLAT TARGET TRANSFORMATION: SIMULATING MULTIPLE SKY POINTING MEASUREMENTS 268

REFERENCES 270

LIST OF ACRONYMS

AD/DD	Architectural and Detailed Design Document
ANX	Ascending Node Crossing
ASCII	American Standard Code for Information Interchange
CAS	CAlibration System
CD	Compact Disk
CFI	Customer Furnished Item
CPU	Central Processing Unit
DLR	Deutschen zentrum für Luft und Raumfahrt
DVD	Digital Versatile Disk
EADS-CASA	European Aeronautic Defence and Space Company- Construcciones Aeronáuticas S.A.
ESA	European Space Agency
ESTAR	Electronically Steered Thinned Array Radiometer
FFT	Fast Fourier Transform
FOM	Figures Of Merit
FOV	Field Of View
GMV	Grupo de Mecánica de Vuelo
GUI	Graphic User Interface
HD	Hard Disk drive
IF	Intermediate Frequency
LNA	Low Noise Amplifier
LICEF	Light and Cost Effective Front-end
LO	Local Oscillator
LTAN	Local Time of Ascending Node
MIRAS	Microwave Imaging Radiometer by Aperture Synthesis
MJD2000	Modified Julian Day of 2000
MSIS-86	Mass Spectrometer-Incoherent Scatter model 1986
NIR	Noise Injection Radiometer
OS	Ocean Salinity
OSF	Orbit Scenario File

PLM	Pay Load Module
PMS	Power Measurement System
RAM	Random Access Memory
RF	Radio Frequency
RFI	Radio Frequency Interference
RISC	Reduced Instruction Set Computer
rms	root mean square
RSC	Redundant Space Calibration
SDF	Swath Definition File
SEPS	SMOS End-to-end Performance Simulator
SM	Soil Moisture
SMOS	Soil Moisture and Ocean Salinity
SSM/I	Special Sensor Microwave / Imager
SSO	Sun-Synchronous Orbit
SUM	SEPS User's Manual
UPC	Universitat Politècnica de Catalunya (Technical University of Catalonia)
UTM	Universe Transverse Mercator projection
WISE	WInd and Salinity Experiment

PREFACE

The present manual is intended to help the user to run the SMOS End-To-End Performance Simulator (SEPS) v4.0. SEPS v4.0 is an upgraded version of SEPS 3.1, which was released in October 2004 and ran under the MATLAB 6.5 environment. SEPS v4.0 runs under both the MATLAB 6.5 and MATLAB 7.0 environments (The Mathworks, Inc.'s MATLAB Releases 13 and 14).

SEPS is a tool to allow the design engineers and the scientists to simulate the performance of the MIRAS instrument aboard the SMOS (Soil Moisture and Ocean Salinity) Earth Explorer opportunity mission. SEPS is based on the existing UPC MIRAS End-to-end Performance Simulator [*Camps*, 1996], and has been implemented for ESA by a consortium led by EADS-CASA, with GMV (orbit propagation, integration and testing), DLR (image reconstruction, only first release of SEPS) and UPC (system performance calculator, brightness temperature generator, receiver modelling, calibration modelling, image reconstruction, cartographic results presentation).

The SEPS v4.0 release includes a number of changes with regard to the SEPS v3.1 software. The changes include: upgrading of the CFI SW version to V3.3; new CFI's features for attitude computation; Sun glint model; back lobe modeling; inclusion of the 4-point calibration method; the flat target transformation method (FTT), the addition of a timeline (decoupling the different calibration and measurement –snapshot- events along the time); and MATLAB v7.0 migration.

The **SEPS software User Manual** is divided into five sections: a brief description of the present document, an introduction to radiometry and synthetic aperture radiometry for users not familiar with these topics, an installation guide, a detailed description of SEPS environment, and two appendices discussing advanced topics on modification of receiver and brightness temperature models.

At its time, the SEPS environment is divided in five parts, which are explained in detail in the Simulator's description (section IV).

- the first one is the **Scenario Management**, where the user can create new scenarios, modifies, renames or deletes existing ones; launch simulations in batch mode; and also define a geographic window containing the time segments for which a selected swath intercepts this geographic window.;
- the second one is the **Scenario Definition**, where the user can modify the orbital parameters, as well as the instrument's ones, the calibration options, the image reconstruction algorithms definition, the image reconstruction algorithm to be applied and finally the simulation timeline, in order to define the scenario to be simulated;
- the third one is the **Mission Simulation**, where the user can finally choose the simulation mode (visibility packets generation, image reconstruction algorithm, or both), and launch single simulations using the parameters defined in the Scenario Definition section;
- the fourth one is the **FOM** (Figures Of Merit) **Simulation Post processing**, which can be used to display graphical results of an already performed simulation;
- and finally, the **FOV** (Field Of View) **Calculator**, used to test the system's performance before a simulation has been run.

SEPS v4.0 has been developed under both the MATLAB version 6.5 (release 13) and MATLAB version 7.0 (release 14) environment to facilitate updates, modifications and the portability of the tool to new releases. MATLAB is a high-performance language for technical computing with large capabilities to perform vector and matrix operations and data visualization. Also, modelling becomes an easier task using MATLAB specific sets of functions.

CHAPTER I. ESA'S SMOS EARTH EXPLORER OPPORTUNITY MISSION

I.1. The SMOS/MIRAS Mission

The European Space Agency (ESA) selected the Soil Moisture and Ocean Salinity (SMOS) mission as the second Earth Explorer Opportunity missions [http://www.cesbio.ups-tlse.fr/us/indexsmos.html; *Silvestrin et al.*, *2001*]. SMOS will be launched in 2007 and its goal is to observe soil moisture (SM) over land and ocean surface salinity (OS), although it will also provide information on root zone SM, vegetation biomass and contribute to research on the cryosphere. SMOS uses the two-dimensional aperture synthesis concept defined in detail during the pre-development of the MIRAS instrument initiated by ESA in 1993 [*Martín-Neira and Goutoule*, *1997*]. This concept is based on a Y-shaped symmetrical antenna array, carrying eighteen identical and equally spaced 1413 MHz antenna/receiver elements in each arm, and fifteen elements (three, plus two redundant ones in each arm) and three redundant noise injection radiometers in the hub. A single receiver chain per antenna element is available, so each unit can only operate on either X or Y polarization in the antenna reference frame at a time. By selecting the antenna polarization properly, it can be operated in dual-polarization or in fully-polarimetric mode [*Martín-Neira and García*, *2000*]. The radiometer calibration includes an on-board calibration by injection of stable noise signals and a vicarious calibration, by periodic imaging of well-monitored and stable regions during the mission. Calibration and imaging issues will be tested in a U-shaped airborne two-dimensional single-polarization interferometric radiometer developed at the Helsinki University of Technology which will be available for land and cryospheric studies [*Rautiainen et al.*, *2000*].

I.2. Basic Concepts on Radiometry

All bodies at a physical temperature different from 0 Kelvin emit electromagnetic radiation. Gases radiate at discrete frequencies corresponding to the difference between atomic energy levels divided by Plank's constant (h). Molecules in solids and liquids are closer than in gases, the energy levels spread, and the radiation becomes a continuum.

When thermodynamic equilibrium is reached, all the energy that is absorbed is then re-radiated (Kirchhoff's law). A "black-body" is an ideal body that absorbs all the incident energy and then in thermodynamic equilibrium re-radiates it following Plank's law. In the microwave region, Plank's law can be approximated by the Rayleigh-Jeans' law: $B_f \approx 2\,k\,T_{ph} / \lambda^2$ [*Ulaby et al.*, *1981*], where B_f is the spectral brightness density [W/Hz/m^2/sr], k is the Boltzmann's constant, T_{ph} is the absolute temperature [Kelvin] and λ is the electromagnetic wavelength.

Real world bodies ("gray-bodies") absorb only a fraction of the incident power, reflecting the rest of it. In thermodynamic equilibrium, the absorbed power is then re-emitted and may present directional properties. The concept of spectral brightness density can be extended to "gray-bodies" by replacing in B_f the physical temperature by the brightness temperature $T_B(\theta,\phi) = e(\theta,\phi)\,T_{ph}$ in the direction (θ,ϕ). The emissivity of the surface $0 \le e(\theta,\phi) \le 1$ is zero for a perfect reflecting material (infinite conductivity), and it is one for a perfect absorber, the "black-body". For an specular surface, the emissivities are simply related to the Fresnel field reflection coefficients at vertical (v) and horizontal (h) polarizations: $e_{\mathrm{v,h}}(\theta) = 1 - \left|\Gamma_{\mathrm{v,h}}(\theta)\right|^2$.

When an ideal antenna is pointing to a "gray-body", the noise power N collected in a bandwidth B small enough to assume that B_f does not change is: $N = k\,T_B(\theta,\phi)\mathrm{B}$ [*Ulaby et al.*, *1981*]. In a real situation, the total noise power collected by a radiometer has other contributions in addition to the electromagnetic radiation emitted by the surface: atmospheric attenuation of T_B, direct or reflected atmospheric emission, and other contributions from outer space. Table 1 summarizes some of the applications and the required frequencies of microwave radiometry. *Ulaby et al.* [*1982*; *1986*] explain in detail the properties and the importance of these particular frequencies.

The thermal emission from a surface bounding a semi-infinite volume at a constant temperature T_{ph}, in the direction (θ,ϕ) is completely described by the Stokes vector [*Yueh et al.*, *1994*]

$$\begin{bmatrix} T_{\mathrm{v}} \\ T_{\mathrm{h}} \\ U \\ V \end{bmatrix} = T_s \begin{bmatrix} e_{\mathrm{v}} \\ e_{\mathrm{h}} \\ e_U \\ e_V \end{bmatrix} = \frac{\lambda^2}{k\eta} \begin{bmatrix} <E_{\mathrm{v}} E_{\mathrm{v}}^*> \\ <E_{\mathrm{h}} E_{\mathrm{h}}^*> \\ 2\,\mathrm{Re}\left(<E_{\mathrm{v}} E_{\mathrm{h}}^*>\right) \\ 2\,\mathrm{Im}\left(<E_{\mathrm{v}} E_{\mathrm{h}}^*>\right) \end{bmatrix} \tag{I.1}$$

where T_{v} and T_{h} are the brightness temperatures at vertical (v) and horizontal (h) polarizations, and U and V are the third and fourth Stokes parameters obtained from the real (Re) and imaginary (Im) parts of the complex cross-correlation of the electric fields measured at v and h polarizations at the antenna position. U and V can also be obtained as the power collected at +45° and -45°, and a left hand circular polarization (LHCP) and right hand circular polarization (RHCP) waves [*Yueh et al.*, *1997*]. T_s is the surface temperature, the symbol < > is the expectation operator, E_{v} and E_{h} are the electric fields at v and h polarizations in the pixel reference frame, η is the wave impedance of the medium and e_{v}, e_{h}, e_U, e_V are the emissivities of the four Stokes elements (the first two Stokes elements are also defined as $I = T_{\mathrm{v}} + T_{\mathrm{h}}$ and $Q = T_{\mathrm{v}} - T_{\mathrm{h}}$ [*Yueh et al.*, *1994*]).

The emissivities of an irregular surface in the XY plane at v and h polarizations, observed from an observation angle θ (relative to 0° at nadir) and an azimuth angle ϕ, are related to its scattering properties by [*Peake*, *1959*; *Yueh* et al., *1994*]

$$\begin{bmatrix} e_{\mathrm{v}}(\theta,\phi) \\ e_{\mathrm{h}}(\theta,\phi) \\ e_U(\theta,\phi) \\ e_V(\theta,\phi) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \frac{1}{4\pi\cos\theta} \iint_{2\pi} \cos\theta_i \begin{bmatrix} \gamma_{\mathrm{vvvv}}(\theta,\phi,\theta_i,\phi_i) + \gamma_{\mathrm{vhvh}}(\theta,\phi,\theta_i,\phi_i) \\ \gamma_{\mathrm{hhhh}}(\theta,\phi,\theta_i,\phi_i) + \gamma_{\mathrm{hvhv}}(\theta,\phi,\theta_i,\phi_i) \\ 2\,\mathrm{Re}\left(\gamma_{\mathrm{vhhh}}(\theta,\phi,\theta_i,\phi_i) + \gamma_{\mathrm{vvhv}}(\theta,\phi,\theta_i,\phi_i)\right) \\ 2\,\mathrm{Im}\left(\gamma_{\mathrm{vhhh}}(\theta,\phi,\theta_i,\phi_i) + \gamma_{\mathrm{vvhv}}(\theta,\phi,\theta_i,\phi_i)\right) \end{bmatrix} d\Omega_i \tag{I.2}$$

where the polarimetric bistatic scattering coefficients $\gamma_{mnpq}(\theta,\phi,\theta_i,\phi_i)$ describe the scattering in the direction (θ,ϕ) of a wave incident from the direction (θ_i,ϕ_i). The coefficient γ_{mnpq} describes the resulting cross-correlation of the scattered waves at m and p polarizations due to incident waves at n and q polarizations respectively (m, n, p, q = h or v).

I.3. Basic Concepts on Aperture Synthesis Radiometry

The results of section I.2 can be extended to the case where the electric fields are measured at different positions. Assuming that the surface under measurement is an extended source of random electromagnetic emission (noise) spatially uncorrelated, that the system is narrow-band, and the source is in the far-field of the array.

In the general case, a correlation radiometer is formed by two antennas located on the XY plane at (x_1,y_1) and (x_2,y_2), and measures the cross-correlation between the signals received by each antenna. This measurement is a sample of the visibility function (V) measured at a spatial frequency $(u_{12},v_{12})=(x_2-x_1,y_2-y_1)/\lambda_0$ (Figure I.1a), being $\lambda_0=c/f_0$ the wavelength at the central frequency of operation (f_0), and can be derived from the Van-Cittert Zernicke theorem [*Goodman*, *1968*; *1985*; *Thompson et al.*, *1986*; *Camps*, *1996*]:

$$V_{12}=V(u_{12},v_{12})\overset{\Delta}{=}\frac{1}{k\sqrt{B_1B_2}\sqrt{G_1G_2}}\cdot\frac{1}{2}\left\langle b_1(t)\,b_2^*(t)\right\rangle=$$
$$=\frac{1}{\sqrt{\Omega_1\Omega_2}}\iint_{\xi^2+\eta^2\le 1}\frac{T_B^{pq}(\xi,\eta)-T_{rec}\delta_{pq}}{\sqrt{1-\xi^2-\eta^2}}F_{n1}(\xi,\eta)F_{n2}^*(\xi,\eta)\tilde{r}_{12}\left(-\frac{u_{12}\xi+v_{12}\eta}{f_0}\right)\exp\left(-j2\pi(u_{12}\xi+v_{12}\eta)\right)d\xi d\eta \tag{I.3}$$

where: $b_{1,2}(t)$ are the analytic signals collected by the antennas at p- and q-polarizations respectively, $(\xi,\eta)=(\sin\theta\cos\phi,\sin\theta\sin\phi)$ are the director cosines with respect to the X and Y axes respectively, $B_{1,2}$ and $G_{1,2}$ are the noise bandwidth and the power gain of the receiving chains (Figure I.1b), $\Omega_{1,2}$ and $F_{n1,2}(\xi,\eta)$ are the equivalent solid angle and the normalized radiation voltage patterns of the antennas, T_{rec} is the physical temperature of the receivers, and $T_B(\xi,\eta)$ [Kelvin] is proportional to $T_B(\xi,\eta)\propto\left\langle E_p^{ant,1}(\xi,\eta)\;E_q^{ant,2\,*}(\xi,\eta)\right\rangle$, where $E_{p,q}^{ant1,2}$ are the electric fields collected by antennas 1 and 2 at p- and q-polarizations, that are related to the electric fields at a given pixel (ξ,η) by geometrical relationships and the co- and cross-polar antenna patterns as defined by Ludwig's third definition [*Ludwig*,

1973; *Claasen and Fung*, *1974*; *Martín-Neira*, *2001*]. In the case of non-zero cross-polar antenna patterns, the following substitutions must be performed in (I.3):

$$
\begin{aligned}
&\left(T_B^{xx}(\xi,\eta)-T_{rec}\right)F_{n1}^{x}(\xi,\eta)F_{n2}^{x*}(\xi,\eta)\rightarrow \\
&\qquad\rightarrow R_{x1}R_{x2}^{*}\left(T_B^{xx}-T_{rec}\right)+\left(R_{x1}C_{x2}^{*}+C_{x1}R_{x2}^{*}\right)\left(T_B^{yx}\right)+C_{x1}C_{x2}^{*}\left(T_B^{yy}-T_{rec}\right) \\
&\left(T_B^{yx}(\xi,\eta)-T_{rec}\right)F_{n1}^{y}(\xi,\eta)F_{n2}^{x*}(\xi,\eta)\rightarrow \\
&\qquad\rightarrow C_{y1}R_{x2}^{*}\left(T_B^{xx}-T_{rec}\right)+\left(R_{y1}R_{x2}^{*}+C_{y1}C_{x2}^{*}\right)\left(T_B^{yx}\right)+R_{y1}C_{x2}^{*}\left(T_B^{yy}-T_{rec}\right) \\
&\left(T_B^{yy}(\xi,\eta)-T_{rec}\right)F_{n1}^{y}(\xi,\eta)F_{n2}^{y*}(\xi,\eta)\rightarrow \\
&\qquad\rightarrow C_{y1}C_{y2}^{*}\left(T_B^{xx}-T_{rec}\right)+\left(R_{y1}C_{y2}^{*}+C_{y1}R_{y2}^{*}\right)\left(T_B^{yx}\right)+R_{y1}R_{y2}^{*}\left(T_B^{yy}-T_{rec}\right)
\end{aligned}
\tag{I.4}
$$

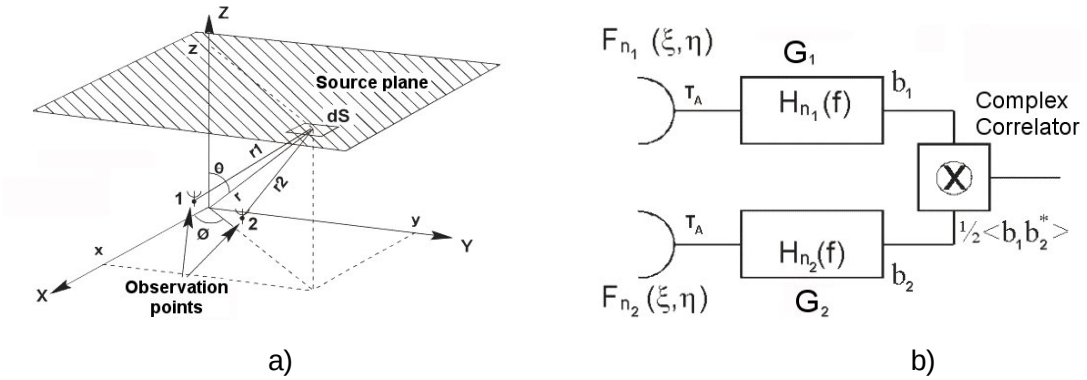
where $R \equiv F_n$ and C are the normalize co- and cross-polar antenna voltage patterns. The relationship between the brightness temperatures in the Earth's reference frame (hh, vv, hv, vh) and the antenna reference frame (pq = {xx, yy, xy, yx}) is given by:

$$
\begin{bmatrix} T_X \\ T_{XY} \\ T_{YX} \\ T_Y \end{bmatrix} = \begin{bmatrix} A^2 & AB & AB & B^2 \\ -AB & A^2 & -B^2 & AB \\ -AB & -B^2 & A^2 & AB \\ B^2 & -AB & -AB & A^2 \end{bmatrix} \begin{bmatrix} T_H \\ T_{HV} \\ T_{VH} \\ T_V \end{bmatrix} \Rightarrow \overline{T}_{B\,antenna} = \overline{\overline{M}}\,\overline{T}_{B\,Earth}\,, \tag{I.5}
$$

being

$$
\begin{aligned}
A &= \cos(\psi - \Phi_{Faraday}); \\
B &= \sin(\psi - \Phi_{Faraday});
\end{aligned}
$$

where $\Phi_{Faraday}$ is the rotation angle through the ionosphere due to Faraday effect, and ψ is a geometrical-related rotation between polarizations.



a) b)

Figure I.1. a) Two antennas of a correlation radiometer. b) Diagram of a correlation radiometer: two receiving chains and a complex correlator.

In (I.3) the function $\tilde{r}_{12}(t)$ is the so-called fringe-washing function, which accounts for spatial decorrelation effects ($t=-(u_{12}\xi+v_{12}\eta)/f_0$) and depends on the normalized frequency response $H_{n1,2}(f)$ of each channel [*Thompson et al.*, *1986*; *Camps*, *1996*; *Camps et al.*, *1999*], and may include several amplifiers, filters and frequency conversions:

$$\tilde{r}_{12}(t)=\frac{e^{-j2\pi f_0 t}}{\sqrt{B_1 B_2}}\int_0^{\infty} H_{n1}(f)H_{n2}^*(f)e^{j2\pi f t}\,df\ ;\ B_{1,2}=\int_0^{\infty}\left|H_{n1,2}(f)\right|^2 df \qquad \text{(I.6)}$$

In an ideal system $\tilde{r}_{12}(\tau)=1$, $H_{n1}(f)=H_{n2}(f)$ and $F_{n1}(\xi,\eta)=F_{n2}(\xi,\eta)$, Equation (I.3) reduces to a continuous Fourier transform between $\left(T(\xi,\eta)-T_{rec}\delta_{pq}\right)\ \left|F_n(\xi,\eta)\right|^2\Big/\left(\Omega\ \sqrt{1-\xi^2-\eta^2}\right)$ and $V(u,v)$.

Particular cases:

i) **Total power radiometer:** If both antennas have the same polarization and are located in the same physical position $(u_{12},v_{12})=(0,0)$ (single-polarization antenna),

then the visibility sample $V_{12}(0,0)$ is equal to the antenna temperature at that polarization.

ii) **Polarimetric radiometer:** If both antennas have orthogonal linear polarizations and are located in the same physical position $(u_{12},v_{12})=(0,0)$ (dual-polarization antenna), then the visibility sample $V_{12}(0,0)=1/2\cdot(U+jV)$, provides a measurement of the third and fourth Stokes parameters.

iii) **Aperture synthesis radiometer:** If both antennas have the same polarization, but are located at different positions then the visibility sample V_{12} is equal to a sample of the two-dimensional Fourier transform of the radiation intensity of the source, which is the result known as the Van Cittert-Zernike theorem [*Goodman*, *1985*]. The first Stokes parameter $I=T_{\mathrm{v}}+T_{\mathrm{h}}$ can be recovered everywhere in the field of view.

iv) **Polarimetric aperture synthesis radiometer:** If both antennas have orthogonal linear polarizations, are located at different positions, and the visibility sample $V_{12}(u_{12},v_{12})$ is measured in any polarization combination, the full Stokes vector can be recovered for any pixel in the field of view.

The MIRAS instrument, payload of the SMOS Earth Explorer Opportunity mission, is a two-dimensional synthetic aperture radiometer that can be operated in modes (iii) or (iv), depending on the selection of the antenna polarization switches [*Martín-Neira and García*, 1999].

I.4. Imaging Considerations in Aperture Synthesis Radiometry

Imaging with real aperture radiometers is achieved by scanning the antenna beam. Radiometer's output provides the measurement at the pixel pointed to by the antenna beam. The resulting image may then be deconvolved to remove errors from secondary lobes and cross-polarization. A review of different imaging configurations can be found in [*Skou*, *1989*]. The most important ones are: cross-track scan with varying incidence angle and pixel size, conical scan of the antenna beam with constant incidence angle and pixel

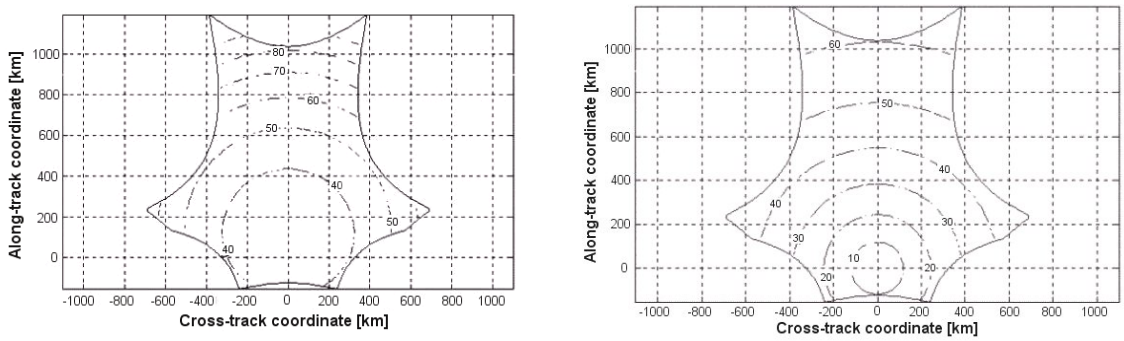
size, and push-broom in which the image is formed by several beams that "sweep" the scene by the movement of the platform. From all of them, the most widely used is the radiometer with a conical scan, a configuration successfully used in the SSM/I series. Along the scan direction, sampling must be at the Nyquist rate, that is twice per pixel, although this requirement can be slightly relaxed depending on the level of aliasing that can be tolerated [*Skou*, *1989*].

In an aperture synthesis radiometer, the visibility function can only be sampled at some spatial frequencies determined by the antenna positions. That is, the reconstructed brightness temperature image will suffer from the artifacts associated with discrete Fourier transforms (aliasing and ringing brightness temperature steps). Applying a window $W(u,v)$ to $V(u,v)$ the side lobes are reduced and the main beam efficiency is improved, at the expense of a widening of the synthetic beam. Depending on the array configuration, aperture synthesis radiometers can be either one- or two-dimensional imagers.

The Electronically Steered Thinned Array Radiometer (ESTAR) was initially proposed by LeVine et al., in the early 1980s and validated by an airborne system developed by the University of Massachusetts at Amherst. It achieves spatial resolution in the along-track direction by the narrow beam of the stick antennas, and in the cross-track direction by aperture synthesis. Aliasing is avoided by setting the antenna spacing to half wavelength.

The Microwave Imaging Radiometer by Aperture Synthesis (MIRAS) aboard the Soil Moisture and Ocean Salinity (SMOS) Earth Explorer Opportunity Mission will achieve two-dimensional spatial resolution by sampling the visibility function in two dimensions with a Y-shaped array. The angular resolution of the instrument is determined by $\Delta u_{\max}$, the maximum distance between (u,v) points. If the visibility samples are not tapered $\Delta\xi_{-3dB} \approx \pi/(2\Delta u_{\max})$. Out of boresight, due to spatial decorrelation (the so-called fringe-washing effects), $\Delta\xi_{-3dB}$ is slightly degraded by a factor $1+\Delta\rho$ in the radial direction, where $\Delta\rho = (B/f_0)\sqrt{\xi_0^2+\eta_0^2}$ [*Bará et al.*, *1998*]. To avoid aliasing the antenna spacing should be $d = 1/\sqrt{3}$ wavelengths [*Mersereau*, *1979*]. If, as in SMOS $d = 0.875 \geq 1/\sqrt{3}$, the Nyquist

criterion is not satisfied and there is aliasing. If one of the Y-arms is oriented perpendicularly to the platform velocity vector, the alias-free field of view in the (ξ,η) domain is limited to the inner pixels of the region defined by the periodic repetition of the unit circle centered at: $\left(\pm 1/\sqrt{3}, \pm 1/d\right)$ and $\left(\pm 2/\sqrt{3}, 0\right)$, which determines which pixels can be recovered without aliasing [*Camps et al.*, *1997a*]. A particular feature of this instrument is that the spatial resolution and incidence angle vary for each pixel in the alias free field of view (Figure I.2), which will help during the geophysical parameter retrieval problem.



a) b)

Figure I.2. Typical alias-free field of view on Earth coordinates for an Y-array, with tilt 32º, orbit height = 755 km and a Y-arm oriented perpendicular to the platform velocity vector: a) spatial resolution and b) incidence angle.

INSTALLATION

This part of the SEPS v4.0 Software User's Manual (SUM) consists of two chapters. ***CHAPTER II*** establishes the basic hardware and software requirements needed to run SEPS v4.0, as well as it makes some further recommendations for enhancing computational speed. ***CHAPTER III*** introduces the user to the basics of the MATLAB environment, including the MATLAB start-up after installation. This chapter also describes the installation process of the SMOS End-to-end Performance Simulator Release 3, as well as its start-up.

CHAPTER II. COMPUTER REQUIREMENTS

This chapter describes the hardware and software requirements for proper SEPS v4.0 simulation processes. The contents of the chapter are

- The first section describes the Operating Systems on which the SEPS v4.0 can be run.
- The second section describes the MATLAB version needed to run the SEPS v4.0, given that successive versions of MATLAB and its technological-specific software packages are not fully compatible. This is of extreme importance to correctly run simulations with SEPS.
- The third section describes the requirements and recommendations on hardware configuration to enhance computational speed when running SEPS v4.0.
- Finally, the last section is dedicated to give some indications to a couple of common questions.

II.1. Operating Systems

The SMOS End-to-end Performance Simulator (SEPS) v4.0 has been intentionally designed to make it available for running in several Operating Systems and hardware configurations. The SEPS v4.0 software includes the Earth Explorer Mission CFIs version 3.3, some considerations must be taken into account under the following Operating Systems:

- Microsoft Windows 2000 for PC –simulations have been run under Windows NT 4.0 and XP without problems-. It is needed the libxml2 version 2.5.7 or later (included in SEPS v4.0).
- UNIX Sun Solaris 2.7 or later version. It is needed the libxml2 version 2.5.7 or later (included in SEPS v4.0).

- Linux 2.4.18 or later version (RedHat 8.0). It is needed the libxml2 version 2.5.7 or later (included in SEPS v4.0).
- Macintosh under MacOS X version 10.2.6 or later. It is needed the libxml2 version 2.5.3 or later (included in SEPS v4.0).

II.2. MATLAB and toolboxes versions

As it has been said above, MATLAB does not guarantee full compatibility between different releases. In fact, compatibility problems with the code generated by the user running under different MATLAB releases are not only possible, but also probable. SEPS v4.0 has been developed for running in both the MATLAB v6.5 and MATLAB v7.0 environments, and therefore the user should ensure to install one of these particular MATLAB releases.

MATLAB is structured as a group of application-specific sets of functions called *toolboxes*. Each of these toolboxes includes the functions to solve most of the problems related to a specific technological area. Some toolboxes are distribution-free and are installed by the MATLAB environment installation program by default, but pay-toolboxes must be specified by the user during the installation process.

During the upgrading process of SEPS for the third release, efforts were made to reduce the number of pay-toolboxes to make its use widely accessible for the SMOS scientific community. **SEPS v4.0 requires ONLY** the following pay-toolbox:

- ***Mapping Toolbox***, version 1.3 (release 13) or version 2.0 (release 14).

The *Mapping toolbox* is needed to project the resulting simulated data on cartographic maps, and therefore if this tool is not to be used its installation can be avoided –however, the user must ensure that the use of the Cartography module is always disabled, as explained in section VII.3.2.

VERY IMPORTANT ⇨ *Using different toolbox versions from the specified above may result in simulation error. This is a MATLAB restriction, since MATLAB does not*

assure full compatibility between successive versions of the same toolbox, neither it does with successive MATLAB releases.

II.3. Hardware recommendations

Due to the high computer performance required by the SMOS End-to-end Performance Simulator (SEPS), the user may consider the following recommendations on the computer configuration to properly simulate SEPS scenarios.

- At least 512 MBytes of RAM for Windows-based systems -otherwise the data storage would be partially performed on the hard disk, dramatically slowing down the simulation speed. The speed in which RAM can be accessed is also important.
- It is recommendable to use a fast hard disk, as there is a wide amount of data loaded from and saved onto HD along the simulation. A 7200-rpm HD is preferable.
- Disk spaced needed: at least 1 GBytes for installing SEPS; however, as temporal and output data take large amounts of disk space, it is strongly recommended to reserve more than 4 GBytes –specially if using the UPC G-matrix image reconstruction algorithm, which easily needs 1.5 additional GBytes-. Also, a large "pagefile.sys" file for virtual memory may be needed for computations (Windows). Please check that your disk space is large enough before running a simulation. This is critical with SEPS 4.0 since a very large number of snap-shots can be simulated in a single run.
- One CD-ROM drives for installation. As large amounts of data are to be returned as outputs of the simulations, the user may consider interesting to have installed a CD or DVD recorder drive, in order to store past simulations while freeing a large amount of hard disk.
- SEPS can only be as fast as the slowest hardware component involved such as a graphic card to display results instantly.
- Using a fast CPU highly improves time simulation. The computer's speed performance depends on the clock speed and the processor family:

Performance = Clock Rate / (# of CPU instructions x cycles per CPU instruction)

Within a processor family the higher the clock rate the better the performance should be; however, one cannot compare processors from different families. For example a RISC microprocessor can execute multiple instructions per clock cycle, and performance can then be better than other processors computing fewer instructions per clock cycle at a higher clock rate.

- Cache memory is fast but expensive and therefore usually limited. The cheaper but slower RAM becomes active whenever the cache memory is exhausted. During SEPS simulations, the use of a powerful cache memory has been proved to speed noticeable the computations. Therefore, taking this item into account is recommended.

Other computer characteristics may not enhance the speed as the user could expect. For example, using more that 1 processor may not enhance speed performance if only MATLAB is running. The reason is MATLAB is not multi-threaded –except for Java-. However, if there is more than one MATLAB user sharing the same machine, then having multiple processors can greatly improve the throughput of the system and the response time of each individual user of MATLAB (and therefore SEPS).

Finally, the user may take into account that sharing CPU time makes applications to be slower. This is particularly true for computational software like MATLAB (and therefore SEPS), strongly depending on the CPU performance.

II.4. Installation Description

II.4.1. Recompiling MEX functions

In case the user wants to recompile the MEX functions (this is the case of **MacOS** users, since MEX files are only provided for Windows, Linux and Solaris), a set of functions is at user disposal. These functions are m-functions that can be used to compile the external libraries.

The SEPSroot/Exec/Dll subdirectories are the place were the code for the MEX functions are located. Each subdirectory contains the code for one MEX-function and an m-function named "compile.m" which is charged of compile the specific library.

This SEPSroot/Exec/Dll subdirectory contains the main function that compiles all the libraries ("compile_all.m"). This function uses the makefile SEPSroot / Exec / Dll / mexopts.sh, which contains the external compiler options to create the MEX-files (for Solaris, Linux and MacOS). For Windows "compile_all.m" asks the users the compiler to be used and to add the appropriate compilation options.

The following compilers are advisable:

- Solaris: gcc/g77 compiler version 3.2.
- Linux: gcc/g77 compiler version 3.3.
- Windows: Microsoft Visual C++ 6.0 Compiler (for C MEX-functions) and Digital Visual Fortran 6.0 (for FORTRAN MEX-functions).
- MacOS: gcc/g77 compiler versión 3.3.

It has to be mentioned that the MEX files generated for this MacOS version will run on any MacOS system, provided that the g77 is installed under /usr/local, and Matlab is installed under /Applications. This is due to the fact that library paths are set into the MEX files and they will crash the simulation if the environment is not correct.

II.4.2. Libxml2 library installation

The addition of the Earth Explorer CFI Software forces to add new considerations during the installation, in case of the user has not previously installed the Earth Explorer Mission CFI.

Due to the *libxml2* is needed for running the Earth Explorer Mission CFIs, it is necessary to define the following environment variables, depending on the platform, during the installation of SEPS (for users which previously have installed the Earth Explorer Mission CFI or the libxml2 library on their computers this step is not necessary provided that the environment variable LD_LIBRARY_PATH is already set to include the location of the existing libxml2 library):

- ***Solaris***:

  In a tcsh shell, "setenv LD_LIBRARY_PATH

  SEPSroot/Exec/Dll/CFIs/LibFiles/Solaris/lib:$LD_LIBRARY_PATH"

  In a bourne shell, "export LD_LIBRARY_PATH=

  SEPSroot/Exec/Dll/CFIs/LibFiles/Solaris/lib:$LD_LIBRARY_PATH"

- ***Linux***:

  In a tcsh shell, "setenv LD_LIBRARY_PATH

  SEPSroot/Exec/Dll/CFIs/LibFiles/Linux/lib:$LD_LIBRARY_PATH"

  In a bourne shell, "export LD_LIBRARY_PATH=

  SEPSroot/Exec/Dll/CFIs/LibFiles/Linux/lib:$LD_LIBRARY_PATH"

- ***Windows***: changes to the system environment are written to the registry, and usually require a restart to become effective. The following steps are needed:

  1. Right-click *My Computer*, and then click *Properties.*
  2. Click the *Advanced* tab.
  3. Click *Environment variables.*
  4. Click the PATH variable, and then click *Edit* to change its value.

5. Add "SEPSroot\Exec\Dll\CFIs\LibFiles\Windows\lib"

- .***MacOS***: Add to environment variable (library-path) the following path SEPSroot/Exec/Dll/CFIs/LibFiles/Macos/lib.

  In a tcsh shell:

  setenv LD_LIBRARY_PATH SEPSroot/Exec/Dll/CFIs/LibFiles/Macos/lib :$LD_LIBRARY_PATH

  setenv DYLD_LIBRARY_PATH $LD_LIBRARY_PATH

  In a bourne shell:

  export LD_LIBRARY_PATH = SEPSroot/Exec/Dll/CFIs/LibFiles/Macos/lib:$LD_LIBRARY_PATH

  export DYLD_LIBRARY_PATH = $LD_LIBRARY_PATH

  Note that in MacOS X Panther (V10.3) the **libxml2** library (version 2.5.4) is already built-in. Panther also includes **libiconv**. Those libraries are usually installed in **/usr/lib/.**

II.4.3. Running the SEPS v4.0 in MacOS

Problems may be faced when trying to run the SEPS v4.0 under MacOS. The following instructions must be followed in order to solve the problems and run the SEPS v4.0 under MacOS X (10.2. and 10.3). The requirements for this particular case are:

1. ***MacOS X Jaguar (10.2)***:
   - Functional Installation of Matlab R13 or R14 including Oroborous (from MATLAB install CD)
   - Developer Environment installed (free download from http://www.apple.com/developer) for 10.2 including gcc 3.3 compiler.
   - g77 v3.4 compiler and libraries for 10.2 (free download from http://hpc.sourceforge.net/g77v3.4-bin.tar.gz)

2. ***MacOS X Panther (10.3)***:

- Functional Installation of Matlab R13 or R14 with X11 from Apple (10.3) including Panther patch (from MATWORKS web site).
- Developer Environment installed (XCode for 10.3) (present on MacOS X Install CD) including gcc 3.3 compiler.
- g77 v3.4 compiler and libraries for 10.3 (free download from http://hpc.sourceforge.net/g77v3.4-panther-bin.tar.gz)

3. ***General***:

- Upgrading of the MEX-functions built from the Earth Explorer CFI version 3.3:
    1. Updating the CFI include files located in "SEPSroot/Exec/Dll/CFIs/LibFiles/Macos/include" with the new files obtained from the Earth Explorer CFI version 3.3: explorer_file_handling.h, explorer_gen_files.h, explorer_lib.h, explorer_orbit.h, explorer_pointing.h and explorer_visibility.h,
    2. Updating the CFI libraries located in "SEPSroot/Exec/Dll/CFIs/LibFiles/Macos/lib" with the new libraries obtained from the Earth Explorer CFI version 3.3: libexplorer_file_handling.a, libexplorer_gen_files.a, libexplorer_lib.a, libexplorer_orbit.a, libexplorer_pointing.a and libexplorer_visibility.a,
    3. Recompiling the MEX files (see Section II.4.1)

II.5. Other questions

1. MATLAB has not been tested under X-Emulation programs. Therefore, running MATLAB (and therefore SEPS) in such environment is not officially supported. If the user experiences problems that appear to be caused by the use of X-Emulation software, the vendor of this software may be contacted directly for additional information, patches, etc…

2. Some Operating Systems have problems for displaying the fonts of the SEPS GUI texts, making them hardly legible. Particularly, this frequently happens in UNIX and Linux. The problem is that the version of the OS lacks certain *TrueType* or *Type1*

fonts, which MATLAB tries to load during its initialization, or that MATLAB is unable to find the existing fonts. Anywhere, the SEPS software can still be used. Some fonts may be difficult to read, but everything should be readable and usable. There are several possible workarounds:

- If a version of XFree prior to version 4.0 is being used, upgrading to version 4.0 or later may be considered.
- Perhaps the *Type1* fonts are installed somewhere else, such as in *usr/share/fonts/default*. If this is the case, then a frequently reported solution is to follow this procedure:
    1. the required fonts must be moved into the X fontpath directory for *Type1*. Then, a utility such as type1inst must be used to recreate the fonts.dir file.
- The "xset +fp" command must be used to add fonts to the fontpath, if the user is working at the machine where MATLAB is running on. Refer to the manual on xset for more information.

However, it is recommended to contact the OS manufacturer in order to solve the problem. It has been also reported that replacing the font.properties file with the MATLAB 6.5 or 7.0 (R13 and R14 respectively) JRE font.properties file solves the problem. However, still some adjustments to the file to remove all the start-up error messages must be made. (Note: This last workaround has not been officially tested by The MathWorks, Inc.).

3. When executing the SEPS on the Linux platform, some errors can occur due to misconfigurations of the MATLAB installation. The solution is to execute the following line before launching MATLAB.

setenv MATLAB_SHELL /bin/sh

CHAPTER III. START-UP OF MATLAB & SEPS v4.0

Once the installation of MATLAB has been completed following the MATLAB installation CD instructions -specifying the toolboxes referred in Chapter II-, the instructions related in the next sections must be executed in order to configure the MATLAB environment for launching SEPS v4.0 simulation sessions.

III.1. MATLAB Environment

Once MATLAB is installed, a MATLAB session can be launched by clicking on the MATLAB icon in the computer's desktop, and then the MATLAB environment is displayed.

The MATLAB environment (MATLAB release 13 has been used to explain the MATLAB environment, this brief description is directly applicable to MATLAB release 14, for more details please see the official MATLAB documentation) consists of several sections, which can be individually deselected to configure the MATLAB environment to fulfill user's needs. Figure III.1 shows the appearance of the MATLAB release 13 environment shown by default:

- the *Command Window* allows the user to type and execute lines of code;
- the *Command History* stores a record of past commands typed in the Command Window;
- the *Current Directory* area shows the files and subdirectories list for the current folder;
- the *Workspace* contains the variables that can be currently managed by the user.

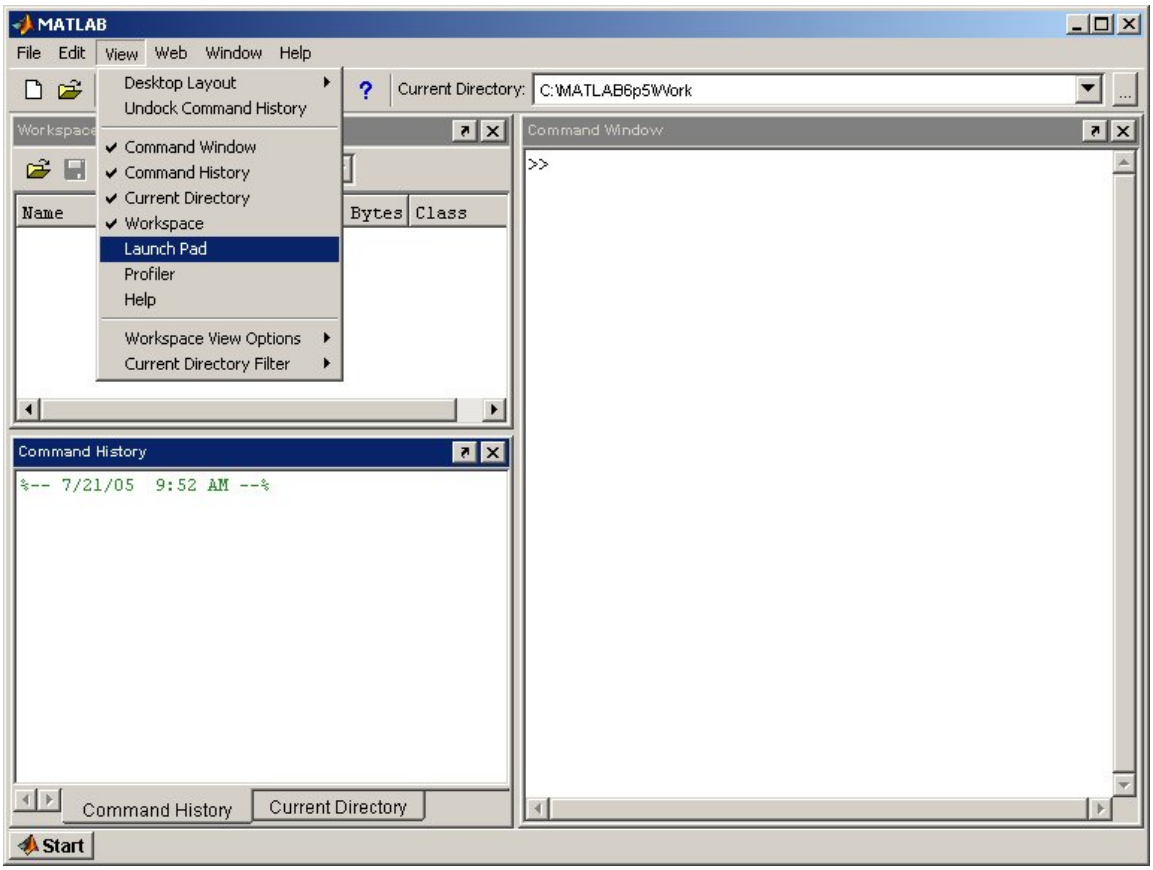


Figure III.1. Default MATLAB environment. Clicking on the View label the configuration of the environment can be changed at user's will.

Other tools are not present by default –such as the *Launch Pad* or the *Profiler*-, but can easily be enabled by the user selecting the ***View*** label and clicking the corresponding option.

III.2. Updating the *Search Path*

The folders where the toolboxes' functions are stored must be included in the *search path*. The *search path* is a list of folders where MATLAB looks for the functions called by the simulator's code. If a function is called and it is not found in one of the folders listed in the *search path* or in the current folder, MATLAB is not able to find it and the simulation is aborted, returning an error message through the MATLAB Command Window.

To avoid this, the user must check that the suitable directories are included in the *search path* and add them to it if they are not.

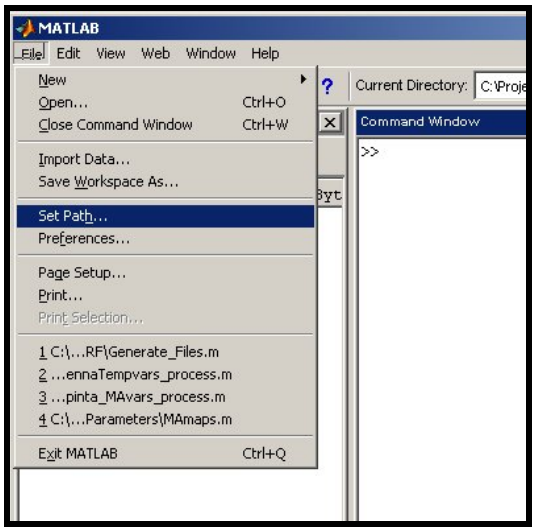


Figure III.2. Clicking the Set Path… *option launches the Path Browser tool.*

The user must click on the ***File*** label of the MATLAB environment and select ***Set Path...*** to display the *Path Browser* tool, where the user has the capability of managing the toolboxes' folders that are needed to simulate the current project:

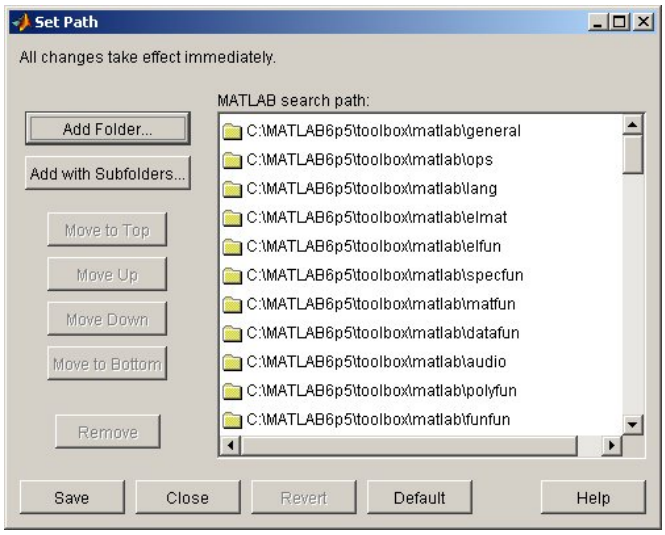


Figure III.3. Path Browser *tool. Clicking* Add Folder… *includes a new folder, while Remove deletes it.*

By clicking ***Add Folder*** / ***Add with Subfolders***, a new directory can be selected through a dialog window and included in the MATLAB *search path*. Pushing ***Remove*** can be used to deselect the unnecessary ones. Apart from the MATLAB free toolboxes installed by default, the following list of MATLAB toolbox' folders must be included in the *search path* when launching a SEPS session:

For both release 13 and release 14:

MATLABroot \ toolbox \ matlab \ general
MATLABroot \ toolbox \ matlab \ ops
MATLABroot \ toolbox \ matlab \ lang
MATLABroot \ toolbox \ matlab \ elmat
MATLABroot \ toolbox \ matlab \ elfun
MATLABroot \ toolbox \ matlab \ specfun
MATLABroot \ toolbox \ matlab \ matfun
MATLABroot \ toolbox \ matlab \ datafun
MATLABroot \ toolbox \ matlab \ polyfun
MATLABroot \ toolbox \ matlab \ funfun
MATLABroot \ toolbox \ matlab \ sparfun
MATLABroot \ toolbox \ matlab \ graph2d
MATLABroot \ toolbox \ matlab \ graph3d
MATLABroot \ toolbox \ matlab \ specgraph
MATLABroot \ toolbox \ matlab \ graphics
MATLABroot \ toolbox \ matlab \ uitools
MATLABroot \ toolbox \ matlab \ strfun
MATLABroot \ toolbox \ matlab \ iofun
MATLABroot \ toolbox \ matlab \ timefun
MATLABroot \ toolbox \ matlab \ datatypes
MATLABroot \ toolbox \ matlab \ verctrl
MATLABroot \ toolbox \ matlab \ winfun
MATLABroot \ toolbox \ matlab \ demos
MATLABroot \ toolbox \ local
MATLABroot \ toolbox \ map \ map

MATLABroot \ toolbox \ map \ mapdisp
MATLABroot \ toolbox \ map \ mapproj
MATLABroot \ work

Only for release 13:
MATLABroot \ toolbox \ matlab \ audio
MATLABroot \ toolbox \ matlab \ winfun \ comcli

Only for release 14:
MATLABroot \ toolbox \ matlab \ scribe
MATLABroot \ toolbox \ matlab \ imagesci
MATLABroot \ toolbox \ matlab \ audiovideo
MATLABroot \ toolbox \ matlab \ codetools
MATLABroot \ toolbox \ matlab \ helptools
MATLABroot \ toolbox \ map \ demos
MATLABroot \ toolbox \ map \ mapformats

where *MATLABroot* must be substituted by the root directory of MATLAB, as set by the user during the MATLAB installation process. Then the user may click on the ***Save*** button to set the definitive path. Finally, the user must click on the ***Close*** button to leave the *Path Browser* tool.

Note 1: *by default, the MATLAB installation process –either installing the MATLAB and the corresponding toolboxes, or adding only specific toolboxes to an already installed MATLAB version- already updates the* search path*. Then, executing the above instructions should not be necessary. They have been specified just to introduce the user to the toolboxes managing.*

Note 2: *the SEPS temporarily deletes from the path the toolbox folders not needed for the SEPS computations, in order to avoid confusions between the names of functions from those toolboxes and functions from the SEPS. However, the original path is turned back on when pressing the* ***Exit*** *button in the SEPS environment. If for any reason the SEPS is closed or aborted without using the Exit button, the temporarily deleted path*

folders will remain unavailable for the current MATLAB session, therefore the use of these toolboxes will be banned. In this case, to recover the original path the user may exit the MATLAB session and open a new one to recover the original path.

III.3. Installing and Configuring SEPS v4.0

The SMOS End-to-end Performance Simulator (SEPS) v4.0 software package is delivered in a CD disk, in a TAR file (and compressed with gzip, generating a .tar.gz file). This format has been chosen due to the fact that files compressed in a TAR-file conserve their permissions, while uncompressed files recorded in a CD are only readable, but not modifiable or suppressible. Otherwise, the user would have to change all the SEPS files permissions –or at least those that can be modified during a simulation-, in order the simulation is not aborted when trying –and failing- to modify one of those files.

The user first must *uncompress* and *untar* the compressed file in the CD on the hard disk drive, reminding to keep the folder structure. In case that the user uses WinZip to uncompress the .tar.gz file, it has been noticed that the default version of this program configuration contains a feature that corrupts some MATLAB binary files. Since this may cause problems uncompressing SEPS, it is recommendable to disable this feature. To do so:

1. go to Menu "Options",
2. submenu "Configuration",
3. tab "Miscellaneous"
4. and de-select "TAR file smart CR/LF conversion".

After decompression on disk, the folders containing SEPS code functions or data must be included in the *search path* –see previous section for further information on this topic-, so as to MATLAB can find them when computing any type of simulation. For this purpose, SEPS includes a function that adds all the needed SEPS folders in the *search path*: the user must type in MATLAB Command Window

>> cd C:/SEPSroot

>> sepsv4

to make SEPS v4.0 root folder the current one (assuming "*C*" is the computer's drive where the software package has been untared, and ***SEPSroot*** is the root folder of SEPS v4.0) and launch a SEPS session, as seen in Figure III.4.

Figure III.4. Appearance of the SEPS v4.0 *after calling a new session from the MATLAB Command Window.*

Calling ***sepsv4*** in the MATLAB Command Window prompt automatically updates de *search path* with all the folders containing the necessary functions and data files to

execute the SEPS v4.0 software package. The user may notice that the *search path* will contain the SEPS needed folders only during the current session. This is useful when using MATLAB for different projects, in order not to mix their functions -which can be named identically only if they cannot be found in the same *search path*-, otherwise one project can erroneously call functions from another project provoking malfunctions. Exiting a MATLAB session and opening a new one creating another *search path* is therefore easier. But if the user needs the MATLAB environment only for running SEPS v4.0, creating the same *search path* in each MATLAB session is inefficient: in that case, the user may open the *Path Browser* tool and click the ***Save*** and ***Close*** buttons to fix the current *search path* permanently.

Anyway, the MATLAB *search path* must include the following folders –apart from the MATLAB default folders for the free-distribution and the pay-toolboxes referred in the previous section-:

SEPSroot / Help /
SEPSroot / Gui / Appearance / Definition /
SEPSroot / Gui / Appearance / Library /
SEPSroot / Gui / Appearance / Figures /
SEPSroot / Gui / Background / Definition /
SEPSroot / Gui / Background / Read-Write /
SEPSroot / Exec / Dll /
SEPSroot / Exec / Dll / Iri95 /
SEPSroot / Exec / Dll / Igrf2000 /
SEPSroot / Exec / S_functions / Antenna / StaticPosition /
SEPSroot / Exec / S_functions / Antenna / DynamicPosition /
SEPSroot / Exec / S_functions / Antenna / Temperature /
SEPSroot / Exec / S_functions / Antenna / AntennaPattern /
SEPSroot / Exec / S_functions / BTGenerator /
SEPSroot / Exec / S_functions / BTGenerator / Additional /
SEPSroot / Exec / S_functions / BTGenerator / GalacticNoise /
SEPSroot / Exec / S_functions / Observables /
SEPSroot / Exec / S_functions / ImageRec / Upc /

SEPSroot / Exec / S_functions / Calibration /
SEPSroot / Exec / S_functions / Receiver /
SEPSroot / Exec / M_functions / SystemCalls /
SEPSroot / Exec / M_functions / Controllers /
SEPSroot / Exec / M_functions / FOMExploitation /
SEPSroot / Exec / M_functions / GeneralPurpose /
SEPSroot / Exec / M_functions / Packets /
SEPSroot / Exec / M_functions / FOMExploitation / Cart /
SEPSroot / Data / Defaults /
SEPSroot / Data / Input / DefaultInputFiles /
SEPSroot / Data / Input / DefaultInputFiles / Receiver /
SEPSroot / Data / Input / DefaultInputFiles / Thermal /
SEPSroot / Data / Input / DefaultInputFiles / Geophysical /
SEPSroot / Data / Input / DefaultInputFiles / Etopo5 /
SEPSroot / Data / Input / DefaultInputFiles / Antennapattern /
SEPSroot / Data / Input / Scenario /
SEPSroot / Data / Output /
SEPSroot / Data / Results /
SEPSroot / Data / Foms /
SEPSroot / Tmp /

III.4. A walk through a basic SEPS v4.0 session

Next paragraphs introduce the user to the most common steps to be followed in a SEPS v4.0 typical session. Although SEPS has been conceived as a versatile set of tools, basic users may follow a standard process when using the SEPS, which is detailed below:

1. Create or load a scenario (see Chapter IV, detailing the *Scenario Management*, and more specifically sections IV.1 and IV.2);
2. Modify some of the instrument configurations, or change one of the files that define the instrument characteristics (see Chapter V, detailing the *Scenario Definition* section, and more specifically section V.2);

3. Compute instrument behaviour, using the *FOV Calculator* tools (see Chapter VIII). Particularly, the basic user may be only interested in the impact of the instrument definition changes on the radiometric accuracy and sensitivity (see then section VIII.1 and VIII.2);
4. The resulting computed values are then displayed (see Chapter VIII, particularly sections VIII.1.2 and VIII.2.2).

VERY IMPORTANT ➔ *If for any reason a SEPS computation crashes, the current folder may not be the adequate one to continue working with SEPS. When launching any computation, the SEPS needs the current folder to be the* ***SEPS root folder****. This is essential because the SEPS finds the suitable files needed for each computation in the database using their relative folder paths. Therefore, if the current folder is not the installation root folder, the SEPS functionalities may be trying to use files from inexistent folders, and then the computations crash.*

SEPS DESCRIPTION

This part of the SEPS v4.0 User's Manual describes the tools provided by this software package. It consists of five chapters, each one illustrating the use of one of the SEPS environment sections:

- Chapter IV describes the tools provided in the *Scenario Management* box. These tools allow the user to *initialize*, *load*, *delete*, *save* or *rename* scenarios; to define geographic areas where time segments, for which a selected swath intercepts these geographic areas, for the current platform's orbit can be found; to launch a chain of successive simulations in *batch mode*; and also to call the SEPS v4.0 *Help* tool.
- Chapter V describes the input parameters menus available through clicking the labels in the lower-left box. This section is called *Scenario Definition* and as its name reveals, allows the user to show input parameter menus in the SEPS environment right box, which can be used to configure all the parameters needed to define a simulation scenario before its start.
- This lower-left box contains also a *Mission Simulation* label, whose description is contained in Chapter VI.
- Chapter VII describes the *FOM* (Figures Of Merit) *Simulation Postprocessing*, which is a set of graphic tools to visually analyze the resulting data obtained from a scenario simulation.
- Finally, Chapter VIII illustrates the *FOV* (Field Of View) *Calculator*, which is a set of radiometric tools to allow the user to evaluate the system's performance, while avoiding launching a complete simulation.

CHAPTER IV. SCENARIO MANAGEMENT

The ***Scenario Management*** box provides the user with some buttons for managing the files containing the data of simulations already performed, and for creating new files to store the data for new simulations to be performed. Two additional buttons allow the user to launch an *automatic scenario* tool to get a timeline file for revisiting a selected geographical zone, and to launch a sequence of simulations in *batch mode*. Finally clicking the last button calls a *Help* tool. Figure IV.1 shows this box, whose options are explained in detail in the next sections.

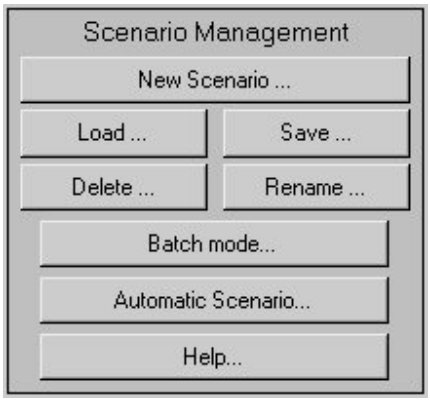


Figure IV.1. Scenario Management *box, used to manage scenario files as well as to create automatic scenarios, to launch simulations in batch mode and to call the* Help *tool*

A scenario is defined by a set of parameters defining the orbit and the instrument itself. These parameters are stored in a file with the ***.sce*** extension in the *SEPSroot\Data\Input\Scenario* folder. When a simulation is run, a folder named after this scenario file is created in the *SEPSroot\Data\Output\Bin* folder, where the results for that simulation are stored.

Next, a description of the use of each of the Scenario Management box' buttons follows.

IV.1. New Scenario

To initialize a new simulation, the user may press the ***New Scenario...*** button. Figure IV.2 show the dialog box that is then opened:

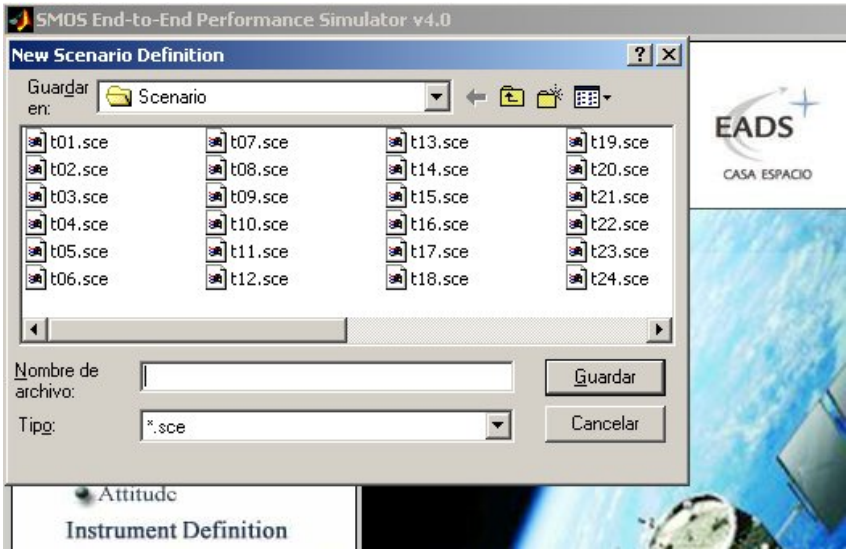


Figure IV.2 Dialog box shown after pushing the New Scenario button

The user may enter an identifier name for the new simulation. Then, the user may press the dialog box' ***Save*** button to set the default SEPS parameters as the input parameters for the new simulation, or the ***Cancel*** button to abort this action. If an existing scenario name is supplied, the dialog box asks the user whether to *cancel* the action or to *overwrite* the existing file. If the *overwrite* option is chosen, all the previous existing information in this scenario file is substituted by the SEPS default parameters. If the cancel option is chosen, the user may enter a new scenario filename.

Note 1*: as SEPS v4.0 includes a number of new capabilities with regard to previous releases, the data structures needed to define the scenario include a number of new input parameters, while excludes some others which are obsolete. Therefore, compatibility between previous SEPS versions defined or simulated scenarios and SEPS v4.0 is not supported, and scenarios need to be redefined and simulated again.*

Note 2: *the user may notice that some messages in this SEPS User's Manual are not written in English. This is due to the fact that some MATLAB functionalities use the Operating System tools –such as dialog windows to select files -. Therefore, the messages displayed by those functionalities are written in the default language of the Operating System.*

IV.2. Load Scenario

To load the input parameters of a previously defined scenario, and to allow the evaluation of the simulation results if this has been already performed, the user may press the ***Load Scenario...*** button. Figure IV.3 shows the dialog box that is then opened:

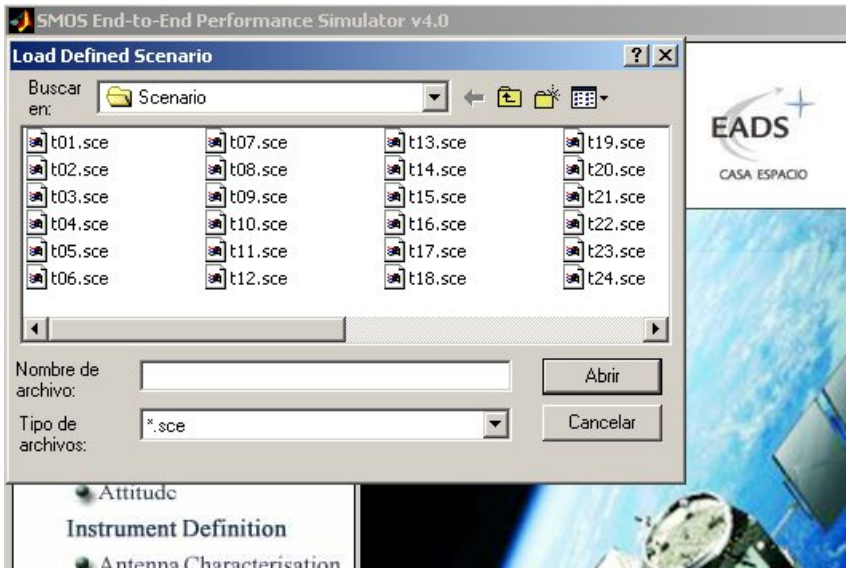


Figure IV.3 Dialog box shown after pushing the Load Scenario *button*

where the user may select an existing scenario filename and press the dialog box's ***Open*** button to perform the loading action, or the ***Cancel*** button to abort it. An error message appears if the scenario filename supplied by the user does not match to any existing scenario file. The user may then supply an existing scenario filename or cancel the loading action.

Note: *the scenario data file may be directly changed by editing the ASCII file, but taking care not to change the format of the file, so the reading environment developed for the simulator will run properly. Any line beginning with a '%' symbol is a comment line, and hence it is no longer considered by the application that reads, writes and decodes this input data files. Any line must begin with a '%' or contain an input parameter, the blank lines are not allowed.*

IV.3. Save Scenario

In order to store the input parameters for an already defined scenario, the user may press the ***Save Scenario...*** button. Figure IV.4 shows the dialog box that is then opened:

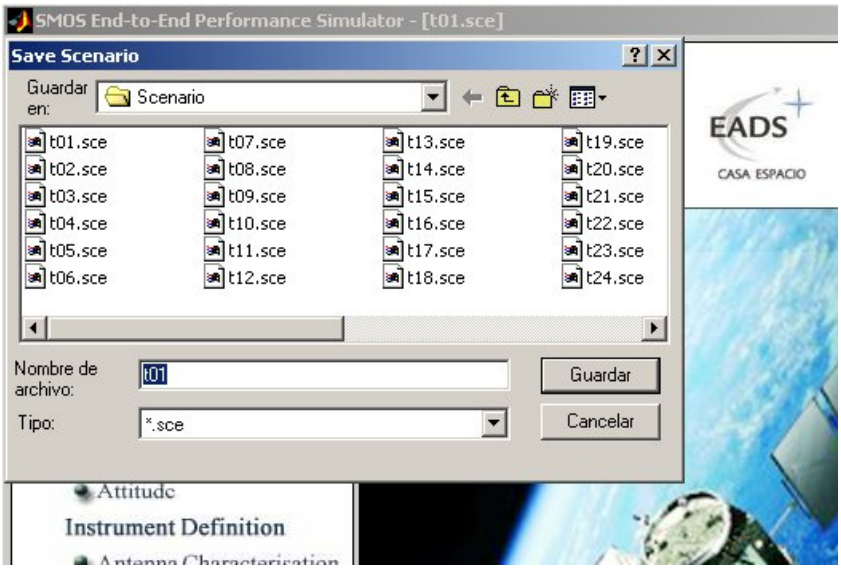


Figure IV.4 Dialog box shown after pushing the Save Scenario *button*

where the user may select or supply a filename for the defined scenario and press the dialog box's ***Save*** button to perform the saving action, or the ***Cancel*** button to abort it. If the scenario filename already exists and the user presses the ***Save*** button, a message asking whether to *overwrite* the existing file or *cancel* the action. The user may press the ***Yes*** button to save, or the ***Cancel*** button to supply another scenario filename.

IV.4. Delete Scenario

To delete a scenario the user may press the ***Delete Scenario...*** button. Figure IV.5 shows the dialog box that is then opened:

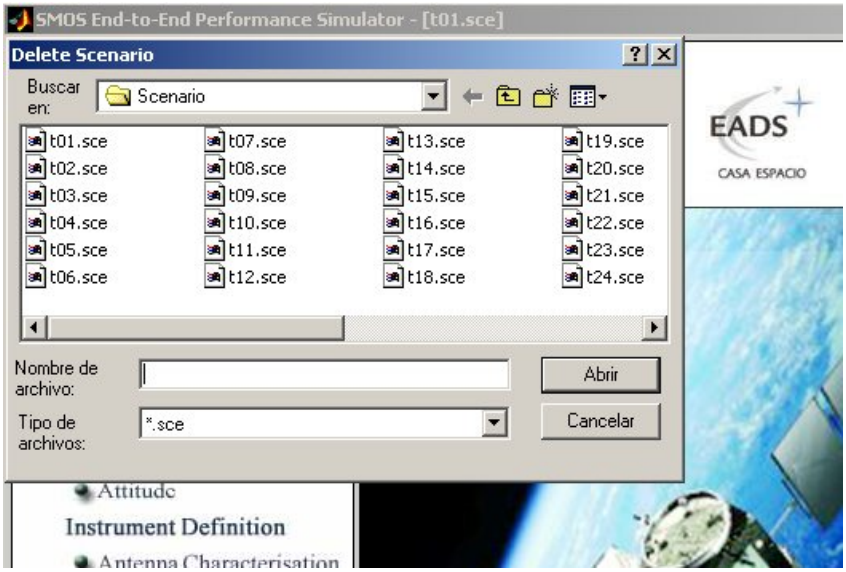


Figure IV.5 Dialog box shown after pushing the Delete Scenario *button*

where the user may select an existing scenario filename and press the dialog box's ***Open*** button to delete it, or the ***Cancel*** button to abort.

IV.5. Rename Scenario

To rename a scenario the user may press the ***Rename Scenario...*** button. A dialog box is then opened where the user may select an existing scenario filename and press the dialog box's **Open** button to continue performing the renaming action, or the ***Cancel*** button to abort it. If the ***Open*** button is pressed, the previous dialog box is closed and the one in figure IV.6 is displayed:

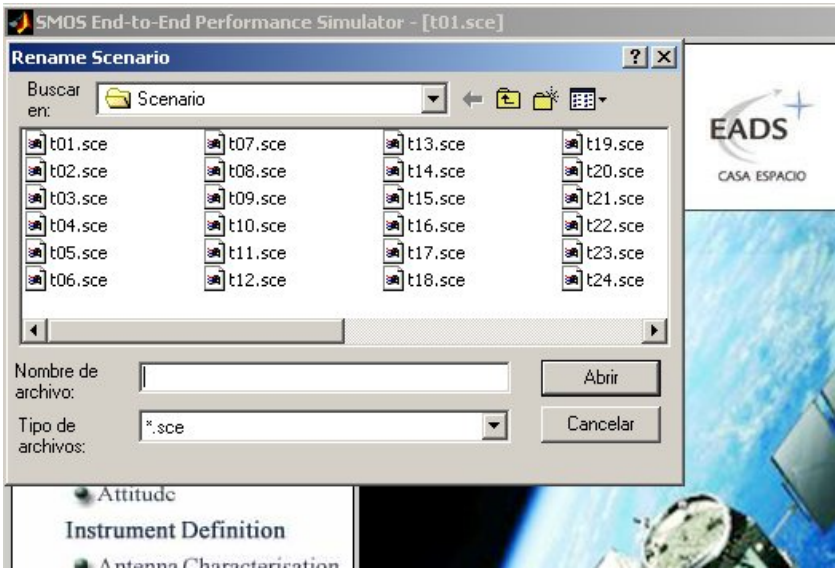


Figure IV.6 Dialog box shown after pushing the Rename Scenario *button and clicking* Yes *when asked for renaming the scenario*

where the user may supply a new filename for the scenario to be renamed. Once the new filename is supplied, the user may press the ***Open*** button to rename, or ***Cancel*** to abort it. If the name supplied matches an existing scenario filename, a message is displayed asking the user whether to *overwrite* the existing scenario file or to *cancel* this action.

IV.6. Automatic Scenario

This is an advanced tool to allow the user select a geographic window where the time segments, for which a selected swath intercepts this geographic window, are computed. This tool is based on the "xv_zone_vis_time" CFI functionality and therefore all the inputs and outputs are predefined by this functionality (for more details see the CFI v3.3 documentation). When the ***Automatic Scenario...*** button is pressed, the interface shown in Figure IV.7 is displayed:

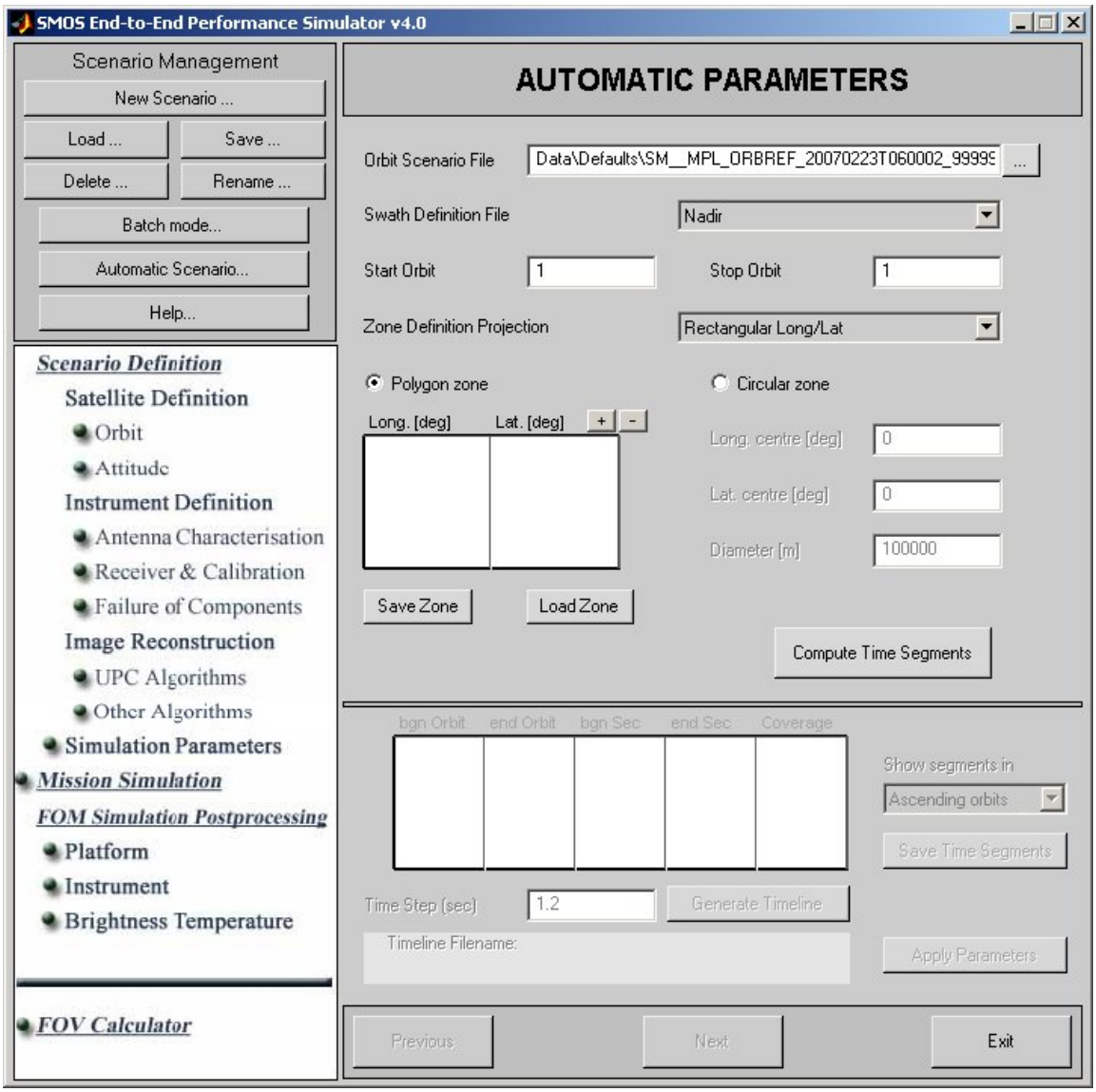


Figure IV.7 Graphic User Interface shown after clicking the Automatic Scenario *button*

The "Automatic Scenario" tool computes all the time segments along the orbit where a predefined swath (part of the Earth surface observed by the instrument) intercepts a user-defined geographic area at the surface of the Earth ellipsoid (see Figure IV-8). Two types of instantaneous swath are considered: line swath or point swath (line or point on the Earth surface at a given time). Note that line and point are simplified views of the MIRAS field of view.

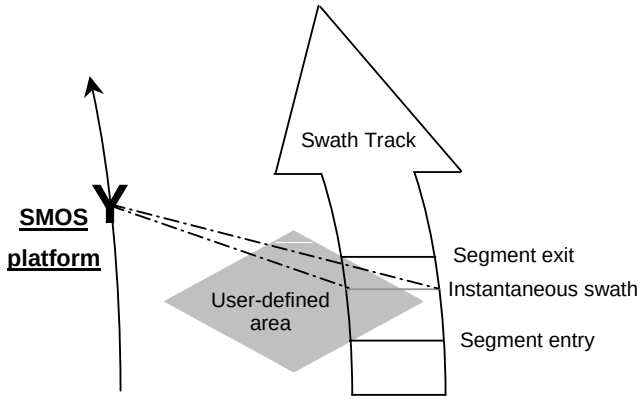


Figure IV.8 Segment definition of the Automatic Scenario Tool

The user may then provide the following inputs:

- ***Orbit Scenario File*** (OSF). This tool performs the orbit propagation using an OSF for the orbit initialization.
- ***Swath Definition File*** (SDF). This file describes the area seen by the instrument along a reference orbit. Three files are available:
    - Nadir. It defines a point-swath (instantaneous swath is a point) pointing to the instrument boresight direction (including the default AOCS attitude parameters and the 32.5º tilt angle).
    - MIRAS P Swath. It defines a line-swath width of 1051 km (instantaneous swath is a line) pointing to the instrument boresight direction (including the default AOCS attitude parameters and the 32.5º tilt angle).
    - MIRAS Q Swath. It defines a line-swath width of 631 km (instantaneous swath is a line) pointing to the instrument boresight direction (including the default AOCS attitude parameters and the 32.5º tilt angle).
- Orbit range: first (***Start Orbit***) and last (***Stop Orbit***) orbit to be propagated related to the OSF file.
- ***Zone definition projection***. Projection used to define polygon sides as straight lines: azimuthal gnomonic (appropriate to high latitudes, where rectangular projections suffers too much distortion), or rectangular latitude/longitude (appropriate to low-medium latitudes).

- There are two radio buttons to allow choosing the type of user-defined zone:
  - ***Polygon zone***: a polygon defined by a set of latitude and longitude points (line is a particular case when the zone is defined by only two points, the line is considered from point 1 to point 2).

    The zone is defined by the area right of the line from point [i] to point [i+1]. The polygon zone can be closed (i.e. the first and the last point are the same) or not. If the zone is not closed, the tool closes it by joining the last point with the first one.

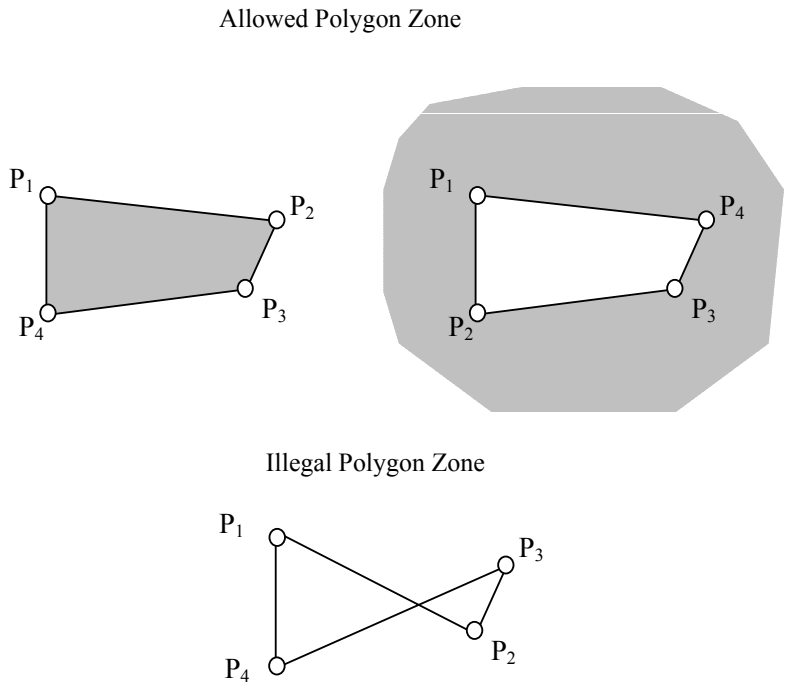


Figure IV.9 Polygon Zone definition

An editable (and scrolling) table provides to user the capability to define and edit polygon zones. Two buttons (on the upper-right corner) allow user to add or remove points from the polygon definition. When user clicks on a cell on the table, the cell changes colour and is selected. When user double-clicks on a cell the editing cursor appears and user can edit the cell contents. To finish editing, the user may press Enter key or click out of the selected cells.

Two additional buttons allow to user save the polygon definition (***Save Zone*** button) to an ASCII file and later reload it (***Load Zone*** button).

- ***Circular zone***: a circle defined by the centre latitude, longitude, and the diameter (point is a particular case for diameter equal to zero). The zone is defined as a circle around the centre point.

Note: *user running on UNIX-based systems may observe a delay in the mouse-related events in SEPS. In fact, the user may need to give some time to the system to react for editing the lat/lon table.*

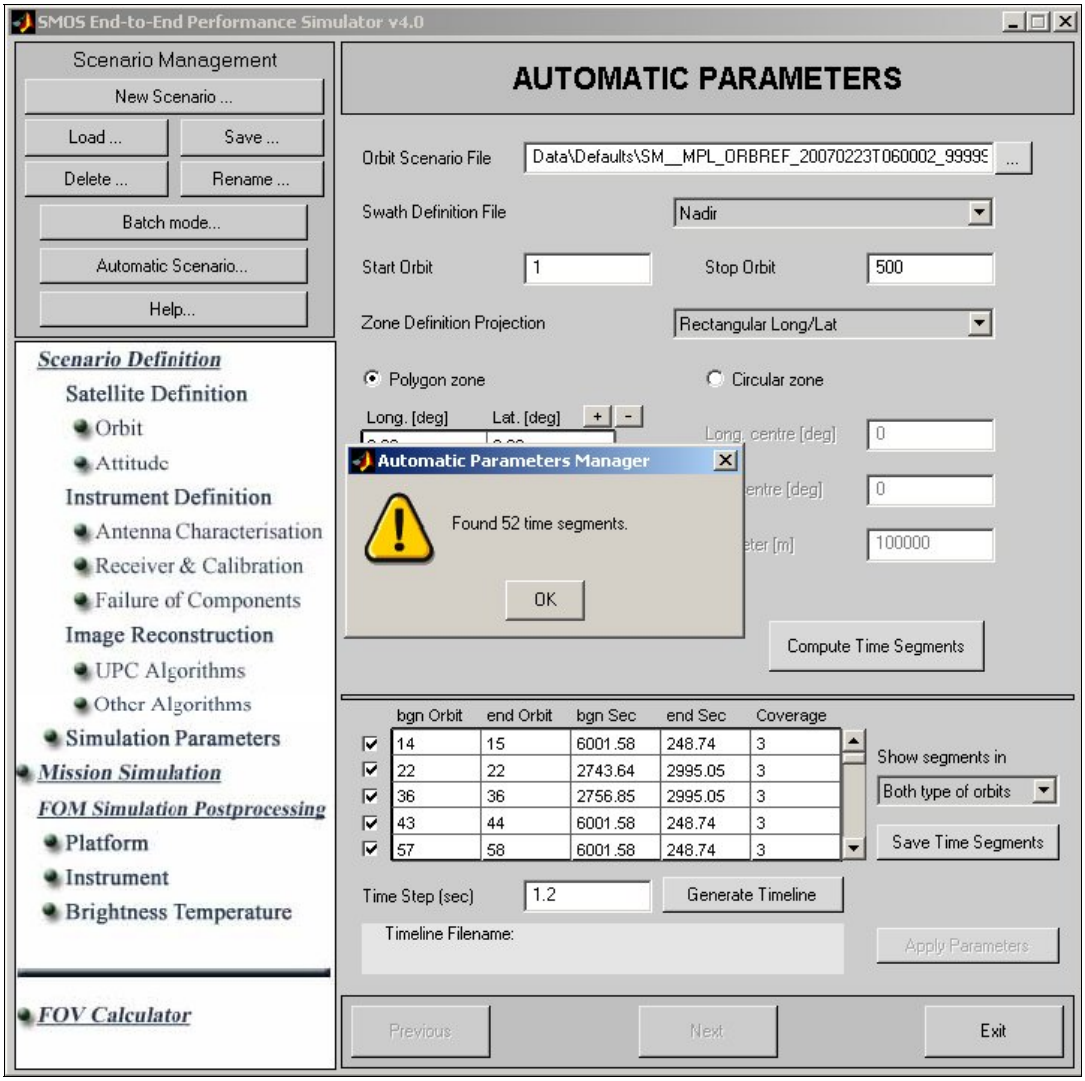


Figure IV.10 Graphic User Interface shown after computing time segments

Clicking the ***Compute Time Segments*** button launches the tool to start iterating and find the time segment in which the selected swath intersects with the user-defined zone. After the execution of the tool a message is displayed, showing the number of time-segments founded along the propagation, and the results are presented on the lower side of the Automatic Scenario panel (see Figure IV.10).

The segments are presented in the lower table sorted chronologically. The following data are shown for each segment:

- Orbit number corresponding to the beginning of the visibility segment.
- Orbit number corresponding to the end of the visibility segment.
- Seconds since the ascending node corresponding to the beginning of the visibility segment.
- Seconds since the ascending node corresponding to the end of the visibility segment.
- Coverage. Zone coverage flag:
    - 0, zone completely covered by swath.
    - 1, zone not completely covered by swath, extending over the left edge of the swath.
    - 2, zone not completely covered by swath, extending over the right edge of the swath.
    - 3, zone not completely covered by swath, extending over both edges of the swath.

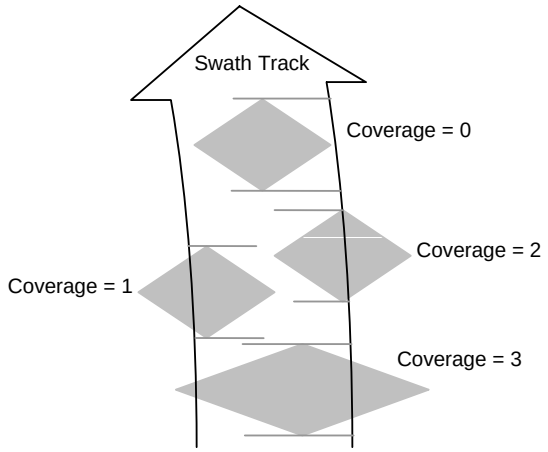


Figure IV.11 Swath coverage definition

The user may press the ***Show segment in*** popup menu in order to select only the ascending, descending orbits, or both types of orbits. Selecting one of these options the check box corresponding to each segment is checked or not as function of the type of orbit in which the time segment occurs. The user can also check the time-segments manually clicking on every check box.

Clicking on the ***Save Time Segment*** button, the time segments obtained from the simulation are saved to ASCII file. A dialog box is opened where the user may select a file and press the dialog box's ***Save*** button to continue saving the time segments action, or the ***Cancel*** button to abort it. In the generated file each row corresponds to a time segment and stores the following information:

- 1st column: orbit number corresponding to the beginning of the segment.
- 2nd column: orbit number corresponding to the end of the segment.
- 3rd column: seconds since the last ascending node crossing corresponding to the beginning of the segment.
- 4th column: seconds since the last ascending node crossing corresponding to the end of the segment.
- 5th column: zone coverage flag for the segment.
- 6th column: seconds since the ANX of the start orbit corresponding to the beginning of the segment.
- 7th column: seconds since the ANX of the start orbit corresponding to the end of the segment.
- 8th column: type of orbit for the segment (1 ascending, -1 descending or 0 both).

The user may provide the time step between snapshots and then press the ***Generate Timeline*** button. A dialog box is opened where the user may select a file and press the dialog box's ***Save*** button to continue performing the generating timeline action, or the ***Cancel*** button to abort it. Once the new filename is supplied and the ***Save*** button is pressed, the previous dialog box is closed and the time-segments are converted into event-record to be simulated by SEPS and save it to ASCII file using the Timeline format

(see Section V.4). The transformation between time-segments and event-record is performed using the following logical sequence:

- For each time segment one time-record is generated.
- The value of the associated check box indicates the "use" flag of the event-record (flag to indicate if the event-record is to be computed).
- The start time of the event-record is equal to the seconds since the ANX of the start orbit corresponding to the beginning of the segment.
- The event-record is composed by events equally spaced (every time step) on the time segment interval (end time minus beginning time).
- Only snapshot events are generated in the timeline file (neither calibration nor G-matrix computation events are considered).

Finally, after the execution of the tool and the generation of the timeline file, the user may press the ***Apply Parameters*** button. This action displays a message, explaining the actions performed and the changes made in the *Scenario Definition* section according to the action previously executed and the obtained results:

Figure IV.12. Message displayed when results obtained from automatic scenario are applied on the current scenario

The *Automatic Scenario* tool updates the following fields of the current scenario with the values selected in the tool: state vector flag is set to orbital propagation from OSF; Orbit Scenario File is set to the selected file; Initial epoch is set to the selected start orbit; and Timeline file is set to the selected timeline file in the tool. When the SEPS simulation for this scenario is run, it uses these parameters if are not specifically changed previously. Additionally, the tool returns to *Scenario Definition* section and its first parameter menu is displayed, where the remaining parameters for the scenario are fixed.

IV.7. Batch Mode

This tool allows the user to launch more than one scenario consecutively. Pressing the ***Batch Mode...*** button, the interface in Figure IV.13 is displayed:

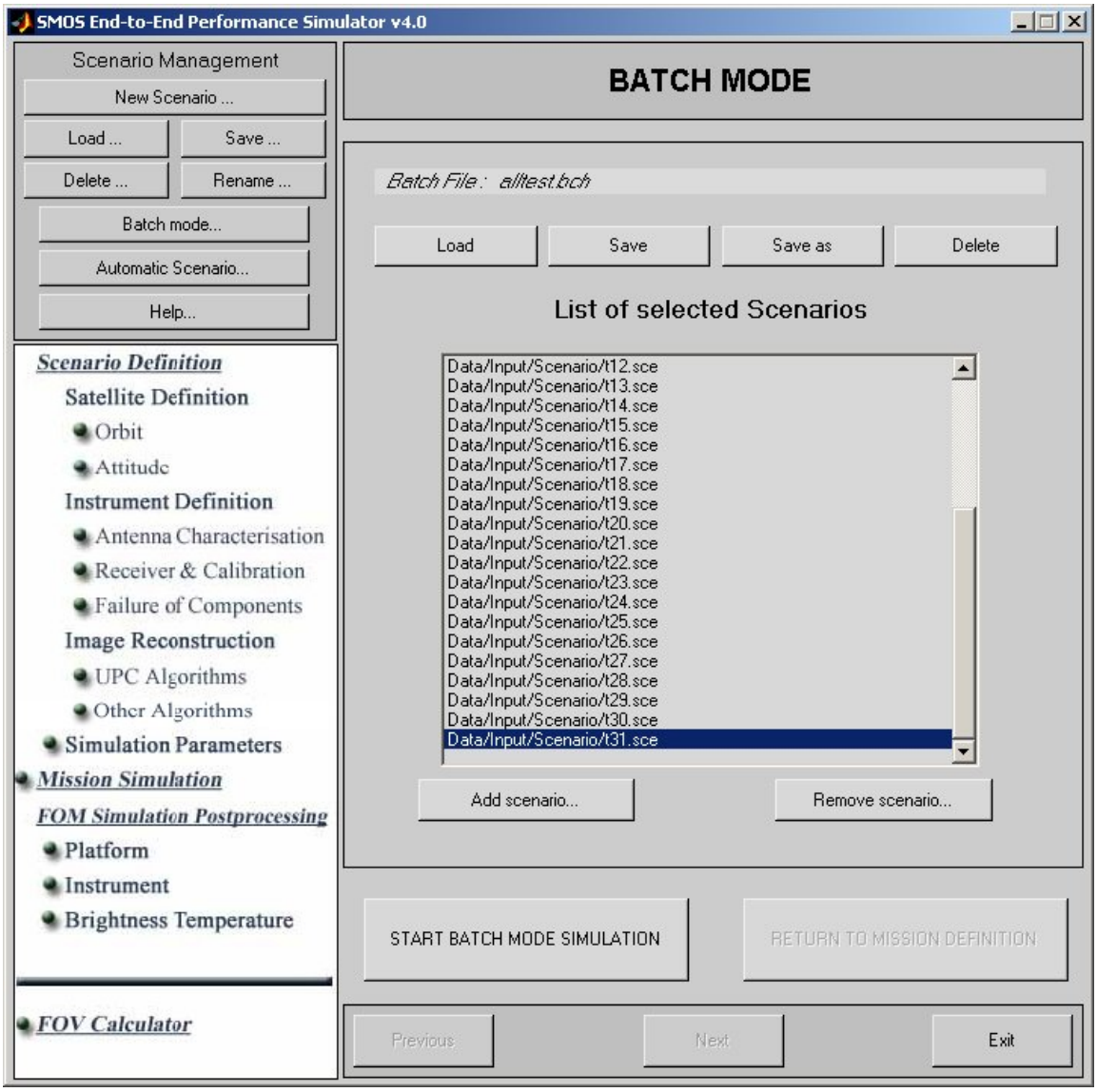


Figure IV.13. Batch Mode *interface for creating and launching a chain of successive simulations*

The box in light grey color lists the scenarios to be run consecutively following this sequence, identified by their paths and filenames with ***.sce*** extension.

The user has the use of four buttons for managing the batch configuration. The user may press the ***Load*** button to load a previously saved batch file (with **.bch** extension), which contains a list of scenario files. A dialog box is opened where the user may select a file and press the dialog box's ***Open*** button to continue performing the loading of the list of scenarios action, or the ***Cancel*** button to abort it. Once the new filename is supplied and the ***Open*** button is pressed, the previous dialog box is closed, the list of scenarios is shown on the list-box and the name of the loaded file is shown in the ***Batch file:*** text-box. The dialog box looks for the batch scenario file in the *SEPSroot / Data / Input / Scenario /* folder.

When a list of scenarios have been configured or edited the user may save it to file pressing the ***Save*** button (overwrite the current batch file) or the ***Save as*** button (save the list in a new batch file). In this last case, a dialog box is opened where the user may select a file and press the dialog box's ***Save*** button to continue performing the saving of the list of scenarios action, or the ***Cancel*** button to abort it. Once the new filename is supplied and the ***Save*** button is pressed, the previous dialog box is closed, the list of scenarios is saved in the file in ASCII format and the new filename is shown in the ***Batch file:*** text-box.

The user may press on the ***Delete*** button in order to remove physically the selected batch scenario file from the hard disk and deselect the selected list of scenarios. This action displays a confirmation dialog (Figure IV.14) to assure the deleting execution.

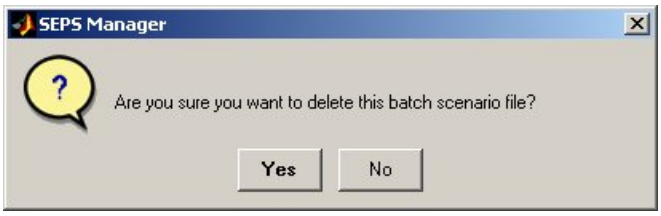


Figure IV.14.Confirmation window for deleting a batch scenario file

The user may press the ***Add scenario...*** button to add the path and filename for another scenario in the list box. Then a dialog box is opened:

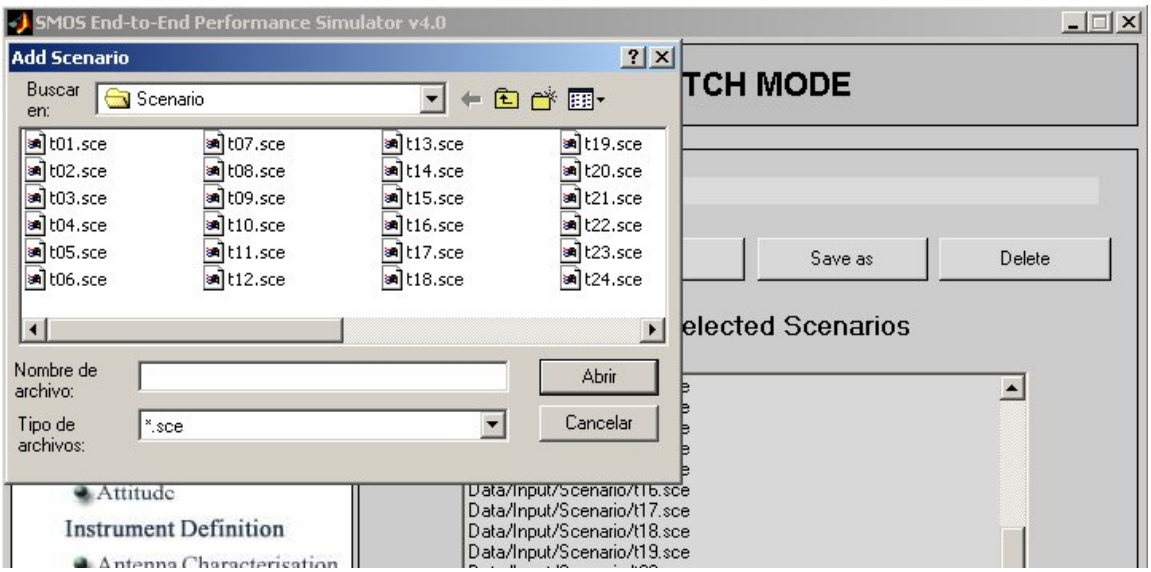


Figure IV.15. Dialog box that appears when the Add Scenario *button is pressed*

where the user may select an existing scenario and then press the ***Open*** button to add it to the batch mode scenario list, or ***Cancel*** to abort the action. If the scenario filename does not exist, an error message is displayed.

Also a ***Remove scenario...*** button is present. The user may select the scenario path-filename string to be removed, and then press the button to eliminate it from the scenario list box:

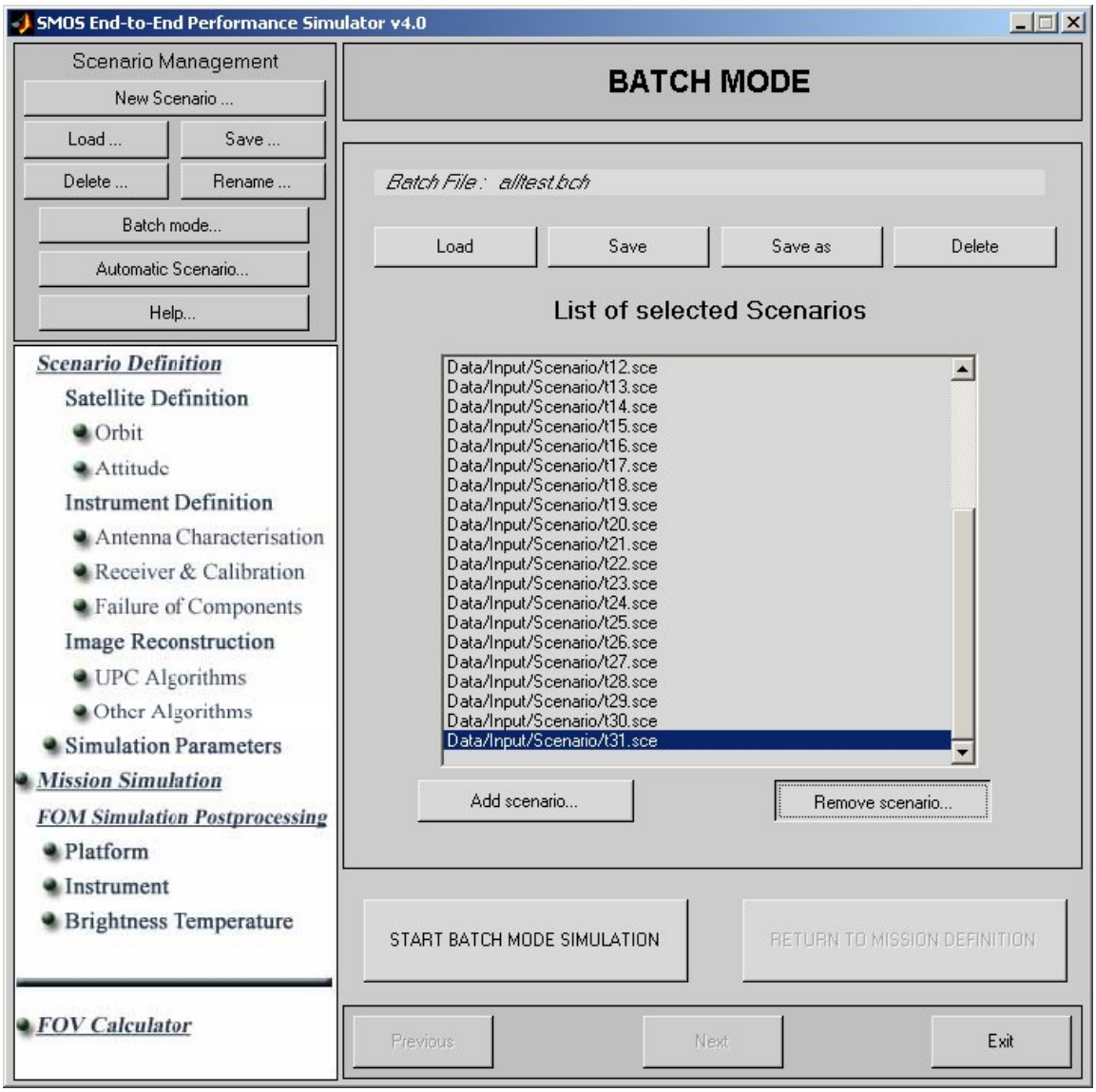


Figure IV.16. Pushing Remove Scenario *when the path of a scenario is selected, deletes this from the chain of simulations to run in batch mode*

When the user completes the list of scenarios to be simulated, then the ***Start Batch Mode Simulation*** button may be pressed to perform a simulation of all the scenarios consecutively, following the order in the light grey box. Either when the simulations are finished or when the idea of launching them is quitted, the user may press

the ***Return to Mission Definition*** button to continue working with SEPS. The button displays the first parameter menu of the *Scenario Definition* section.

IV.8. Help

The last button in the *Scenario Management* box is the ***Help*** button. By pressing it the user accesses a review of the functionalities of SEPS v4.0, as it is shown in Figure IV.17.

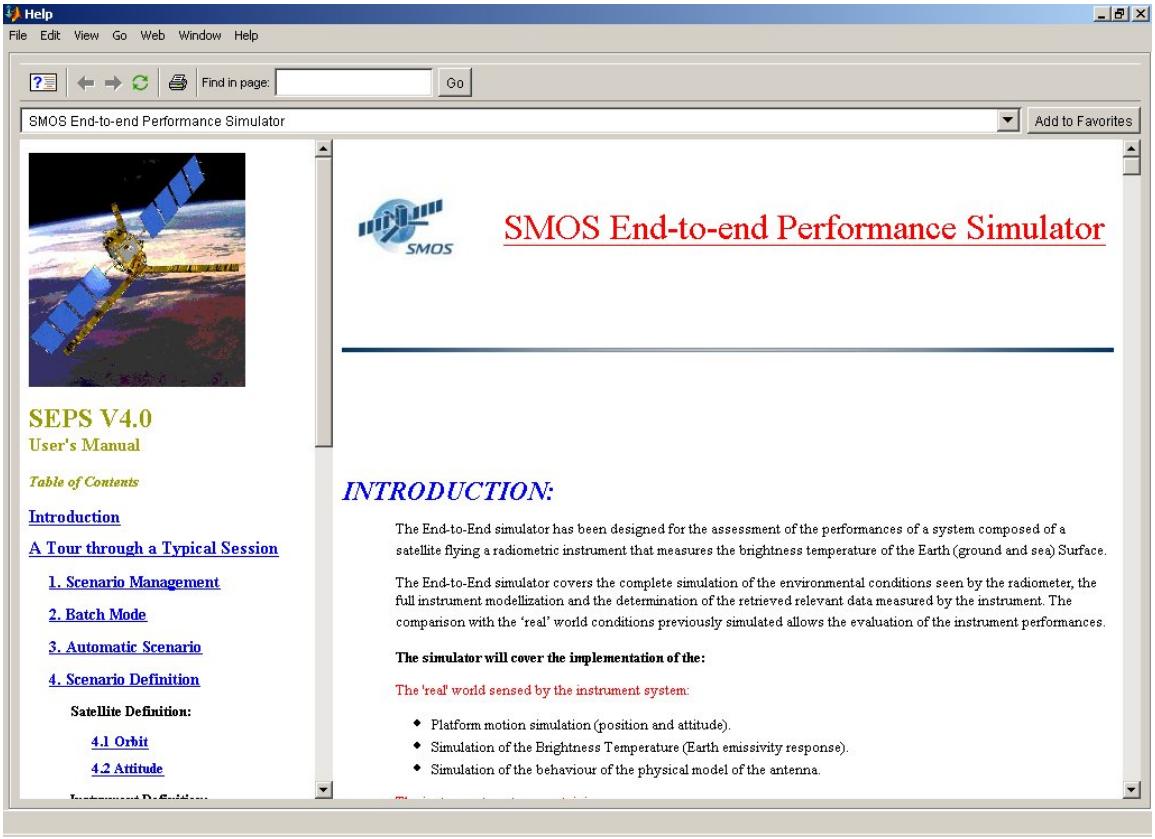


Figure IV.17. Figure shown when Help *button is pushed*

CHAPTER V. SCENARIO DEFINITION

This chapter describes the *Scenario Definition* section of the SEPS environment, which consists of a set of labels calling menus where the input parameters needed to define a scenario can be updated. Once a new scenario is initialized, or an existing scenario is loaded, the user may change the parameters before launching a simulation.

The chapter is organized in four sections, corresponding to each one of the Scenario Definition sections:

- *Satellite Definition*, where the orbital and attitude parameters for the current scenario can be updated;
- *Instrument Definition*, calling the input menus to define the instrument's design, such as the antenna characterization or the receiver and calibration parameters;
- *Image Reconstruction*, where the user can modify the characteristics of the image reconstruction algorithms available in SEPS v4.0;
- *Simulation Parameters*, where simulation parameters are assigned (e.g. timeline, sun and moon effects, type of image reconstruction method to be used, etc.).

V.1. Satellite Definition

In this section the parameters needed to define the orbital module computation are described. The following pages detail the input parameters to this module. For further information on this topic the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, sections 6.7 to 6.10).

V.1.1. Orbit

The input parameters to the orbital propagation module are fixed using the *Orbit* menu in the *Satellite Definition* subsection. Figure V.1 shows the appearance of this menu when called clicking the ***Orbit*** label.

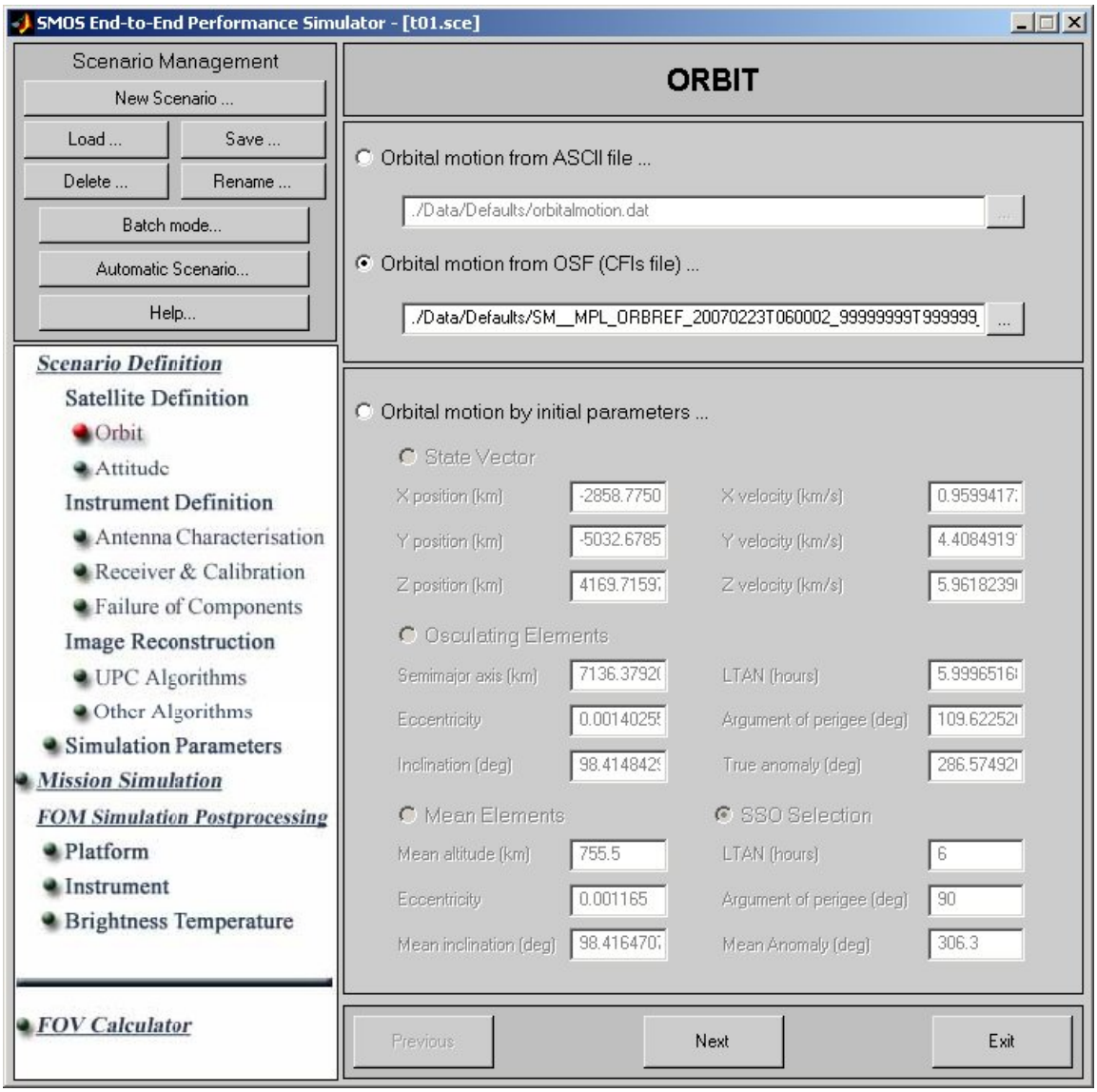


Figure V.1. Default appearance of the Orbit *menu*

The ***Orbit Parameters*** interface allows the user to configure the orbit, choosing one of three mutually exclusive alternatives: either selecting a file where platform's positions, velocities and times are contained; obtaining the initial propagation parameters form a OSF file (CFI format), or using an interface to update the values individually.

By default the ***Orbital motion from OSF file (CFIs file)…*** option is selected. This option allows the user to examine the directory tree and look for an OSF (for more details about the OSF files please see the CFI documentation). The user can provide a new file by clicking on the push button at the right side of the box, which opens a dialog box for looking a new OSF file, as in the example seen in Figure V.2.

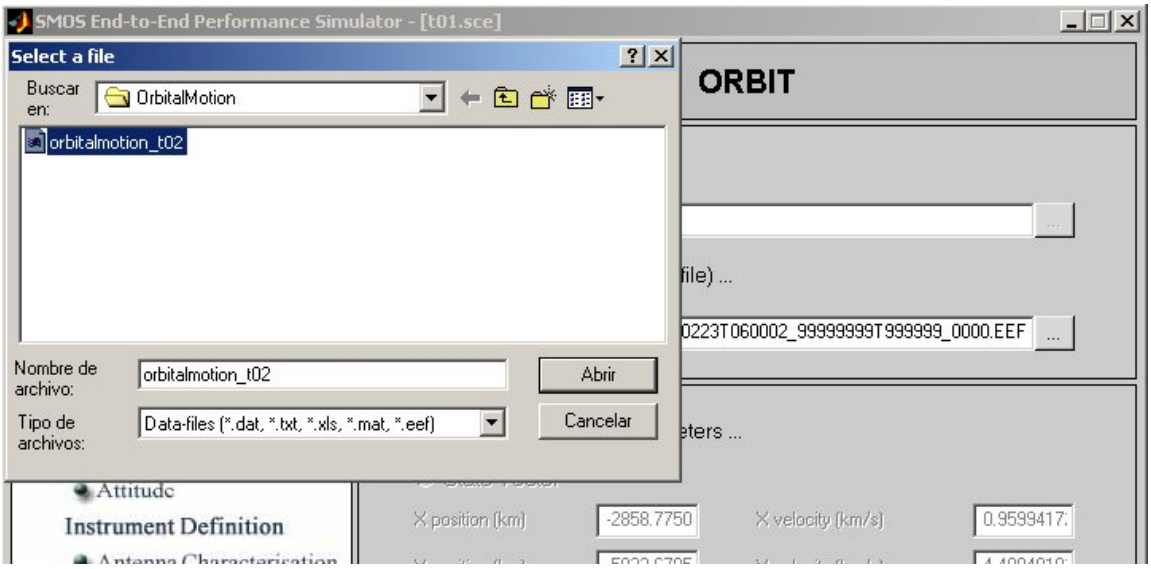


Figure V.2. Dialog box opened when clicking the right side … button, to choose the OSF file

The dialog box looks for the OSF files in the *SEPSroot / Data / Input / UserInputFiles / OrbitalMotion /* folder. Anyway, files in other folders can also be selected. The user may notice that the default OSF file, containing the default SMOS orbit or *baseline orbit* and named as:

SM__MPL_ORBREF_20070223T060002_99999999T999999_0000.EEF

, is not present in the referred folder but in the *SEPSroot / Data / Defaults /* folder.

Other alternative is that the user defines the orbital motion supplying a user's orbital motion file (ASCII file). To enable this option, the user must click on the ***Orbital***

motion from ASCII file … radio button. Defining the orbital motion from ASCII file allows the user to examine the directory tree and look for a file containing the platform's position, velocity, acceleration and time of the simulation. The user can provide a new file by clicking on the push button at the right side of the box, in a similar way as it has been explained for the case of OSF file.

The format required for a user-supplied orbital motion file must consist of a number of rows –containing data for successive platform's positions in time-, each one with 10 columns containing the following values:

- Simulation epoch, in MJD2000;
- X /Y /Z components of the platform's position in the Geocentric Mean of 2000 CS reference frame, in meters;
- X /Y /Z components of the platform's velocity in the Geocentric Mean of 2000 CS reference frame, in meters/second.
- X /Y /Z components of the platform's acceleration in the Geocentric Mean of 2000 CS reference frame, in meters/second^2.

Please refer to the default file *orbitalmotion.dat* in *SEPSroot / Data / Defaults* to learn more on the format needed for the orbital motion files.

The last alternative is that the user defines the orbital motion supplying its initial parameters. To enable this option, the user must click on the ***Orbital motion by initial parameters…*** radio button. Three alternatives are offered to the user for defining the orbital motion by its initial parameters: the *State Vector*, the *Mean Elements* and the *Osculating Elements* definitions. Changing the initial parameters using one of these three alternatives, results in the automatic change of the values of the other alternatives. Figure V.3 shows the appearance of the resulting interface, with the *State Vector* option enabled by default.

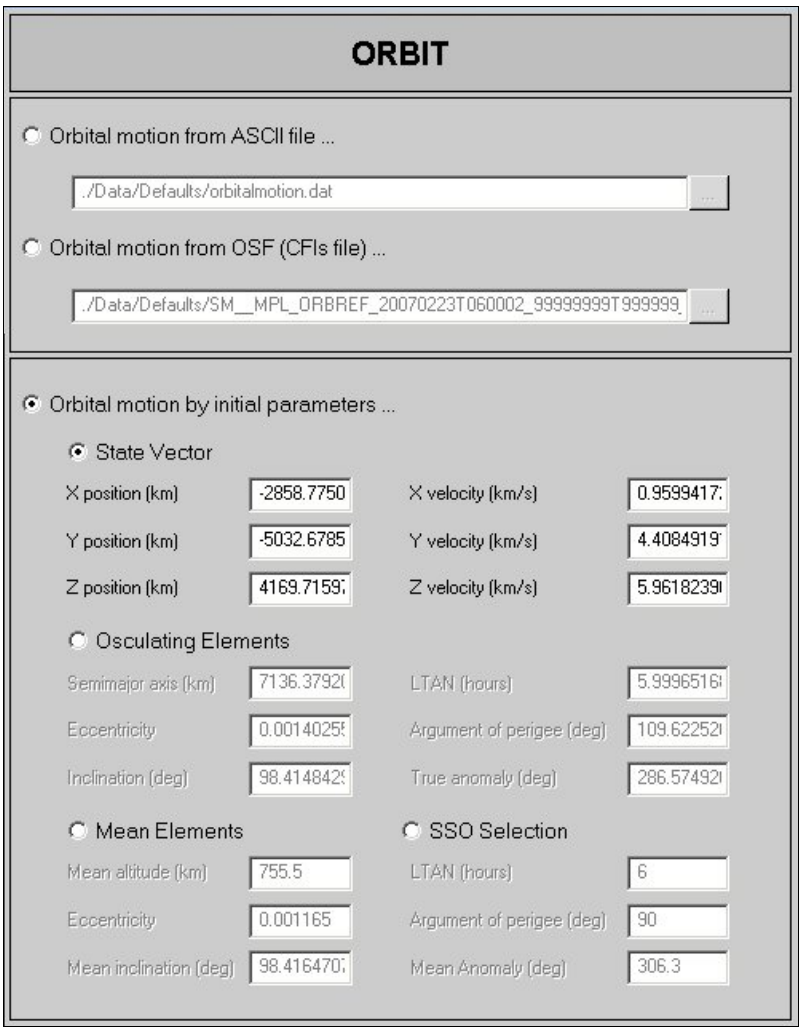


Figure V.3. Appearance of the Orbit Parameters *submenu after choosing the* Orbital motion by initial parameters… *option*

As it is seen in the figure above, the update of the parameters is performed through the use of edit boxes. The user may type the new values for the ***State Vector*** alternative:

- *X/Y/Z position*: Cartesian components of the initial position of the platform in the Geocentric Mean of 2000 CS reference frame, in km.
- *X/Y/Z velocity*: Cartesian components of the initial velocity of the platform in the Geocentric Mean of 2000 CS reference frame, in km/s.

The second alternative available is the use of the ***Osculating Elements*** to define the initial parameters of the orbital motion. Here also the reference frame is defined by the Geocentric Mean of 2000 CS. The following parameters may be changed by the user if needed to:

- *Semi-major axis*: half of the ellipse's major axis of the platform's orbit, in km.
- *Eccentricity*: elongation of the platform's elliptic orbit, between 0 and 1.
- *Inclination*: elevation of the platform's orbit above the Equator, in sexagesimal degrees.
- *LTAN*: local time of the ascending node of the platform's orbit, in hours.
- *Argument of perigee*: argument of the point of the platform's orbit closest to the Earth, in sexagesimal degrees.
- *True Anomaly*: polar angle of the platform in a Kepler orbit, measured in the orbital plane from the position at closest approach, in sexagesimal degrees.

The third alternative is to use the ***Mean Elements*** to define the initial parameters of the orbital motion. Here also the reference frame is defined by the Geocentric Mean of 2000 CS. The platform's orbit classical mean parameters that can be updated by the user through this option are:

- *Mean altitude*: the mean height of the satellite above the Earth's surface, in km.
- *Eccentricity*: elongation of the platform's elliptic orbit, between 0 and 1.
- *Mean inclination*: mean elevation of the platform's orbit above the Equator, in sexagesimal degrees.
- *LTAN*: local time of the ascending node of the platform's orbit, in hours.
- *Argument of perigee*: argument of the point of the platform's orbit closest to the Earth, in sexagesimal degrees.
- *Mean Anomaly*: mean polar angle of the platform in a Kepler orbit, in sexagesimal degrees.

This last alternative provides the possibility of forcing the orbit to be sun-synchronous (SSO). If the user enables the ***SSO Selection*** radio button, the edit box of the mean inclination parameter becomes disabled, as it can be observed in Figure V.4.

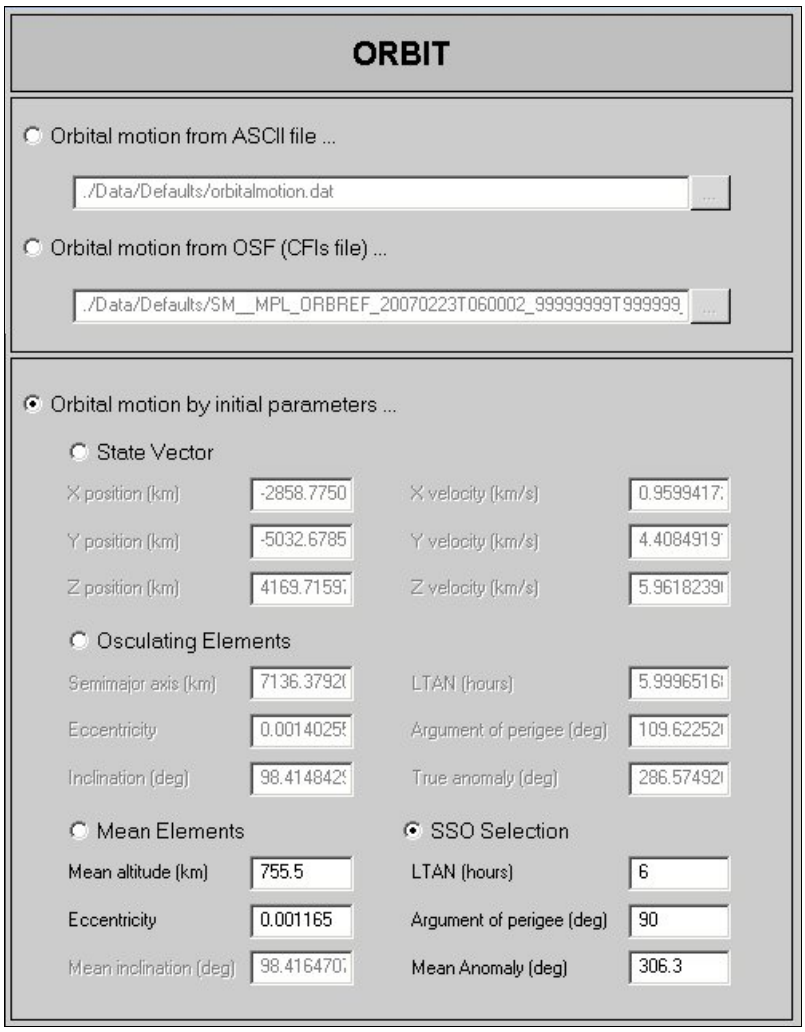


Figure V.4. Mean Elements *alternative for providing the orbital motion initial parameters, when* Sun-Synchronous Orbit (SSO) *is chosen*

For further information on the orbital motion characterization, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.7).

***NOTE**: for long-term simulations, it is recommended to use an OSF file to perform the orbit propagation in order to not lose accuracy in the propagation.*

V.1.2. Attitude Definition

To define the spacecraft attitude, the user may click the **Attitude Definition** label of the *Satellite Definition* section in the *Scenario Definition* box. The attitude can be defined either using a file containing the attitude profile or using the interface provided to enter the data manually. Figure V.5 shows the default appearance of the Attitude Definition menu.

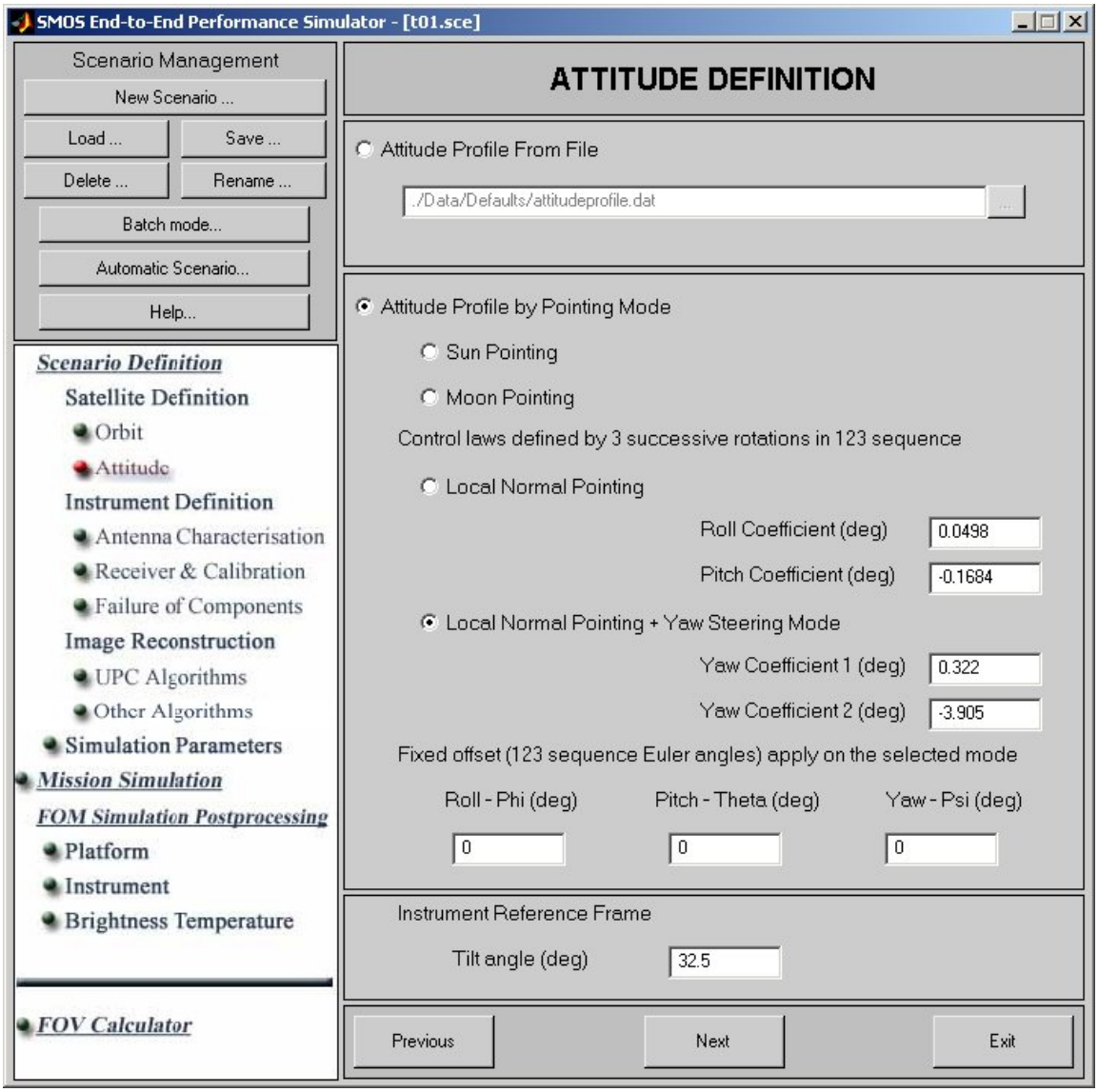


Figure V.5. Default appearance of the Attitude Definition *menu*

If the ***Attitude Profile from File*** radio button is activated, the user must examine the directory tree of the computer and look for the appropriate file containing the attitude data. As it can be seen in Figure V.5, the default file provided by the SEPS v4.0 is stored in the *SEPSroot / Data / Defaults /* directory. The user can provide a new file by clicking on the right side ***…*** pushbutton, which opens a dialog box for looking a new attitude definition file, as in the example seen in Figure V.6. The dialog box looks for the new user's attitude file in the *SEPSroot / Data / Input / UserInputFiles / AttitudeProfile /* folder.

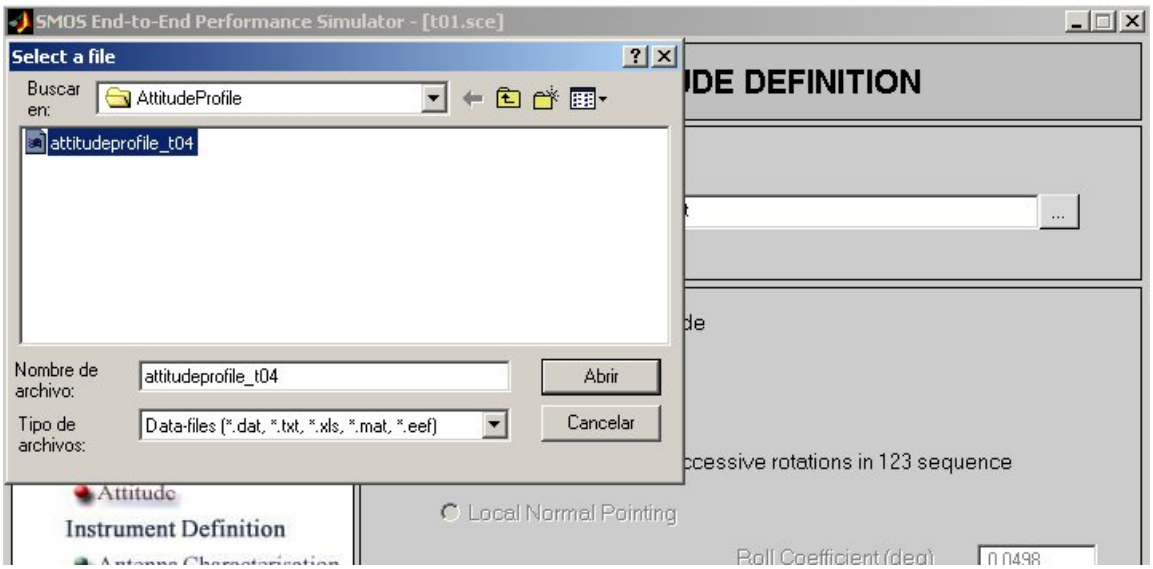


Figure V.6. Dialog box opened when clicking the right side … button, to choose the attitude profile *file*

The attitude file must contain a number of rows with the attitude parameters' values for each step of the simulation, in four columns following the format:

- Simulation epoch, in MJD2000;
- Attitude profile in Euler angles, in the rotation sequence 1-2-3 (roll, pitch and yaw from Geocentric Mean of 2000 CS to Body CS), in sexagesimal degrees.

Please refer to the default file *attitudeprofile.dat* in *SEPSroot / Data / Defaults /* to know more on the format needed for the user's attitude file.

The alternative is that the user defines the attitude profile supplying the values of the needed parameters through the interface that is enabled when clicking ***Attitude Profile by Pointing Mode***. Alike, this option offers four different alternatives, depending on the pointing selection:

- *Sun Pointing*: the user must select this one to describe the attitude referred to the Sun direction, clicking the corresponding radio button. This option is available in order to perform calibrations using the Sun as natural source. Units are sexagesimal degrees.
- *Moon Pointing*: to describe the attitude referred to the Moon direction, click this radio button. This option is available in order to perform calibrations using the Moon as natural source. Units are sexagesimal degrees.

The last two alternatives allow the user to define the attitude through the *control laws defined by 3 successive rotations in 123 sequence*:

- *Local Normal Pointing*: the *Roll* and *Pitch* coefficients given in the edit boxes are used to rotate around *X* and *Y* (local orbital frame), in the following way:

  $$roll = cx * \sin(\omega t)$$

  $$pitch = cy * \sin(2 * \omega t)$$

  with cx and cy being the roll and pitch coefficients and ωt being the true latitude

  Units are sexagesimal degrees.
- *Local Normal Pointing +Yaw Steering Mode*: the *Roll* and *Pitch* coefficients given in the edit boxes are used to rotate around *X* and *Y* (local orbital frame); and two coefficients supplied by the user through the edit boxes at the bottom of the *Attitude Definition* interface define the *Yaw* angle, which is used to rotate around *Z* (local normal frame), in the following way:

  $$roll = cx * \sin(\omega t)$$

  $$pitch = cy * \sin(2 * \omega t)$$

  $$yaw = cz_1 * \sin(\omega t) + cz_2 * \cos(\omega t)$$

  with cx, cy, cz_1 and cz_2 being the roll, pitch and first and second yaw coefficients respectively and ωt being the true latitude

Units are sexagesimal degrees.

The above four alternatives define the attitude using an *offset defined by Euler angles in 123 sequence* (roll, pitch and yaw), this meaning that the reference point is the one selected by the user (Sun, Moon, Local Normal or Local Normal + Yaw Steering), and the angles' values supplied in the edit boxes are referenced to that direction to achieve the spacecraft's pointing.

Finally, the user may be interested in using a combination of one of the Pointing modes and the Tilt Angle parameter. The tilt angle defines the pitch rotation between the Body CS and the Instrument CS. This makes the array boresight to point in the body Z-axis, tilting the array around the body Y-axis (pitch) by an angle given by the value supplied by the user in the ***Tilt Angle*** edit-box. By default is 32.5°, units are sexagesimal degrees.

Further information on this topic is provided in the AD/DD document (SO-TN-GMV-PLM-0003, section 6.9), supplied with the SEPS v4.0 software.

V.2. Instrument Definition

In this section the parameters needed to define the SMOS/MIRAS instrument are described. The following pages describe the interfaces through which the parameters affecting the calibration procedure, the array and the antennas characterization, and the failure of the receivers and correlators, can be modified by the user. For further information on these topics, please refer to the AD/DD document.

V.2.1. Antenna Characterization

The definition of the array is performed through two interfaces, one to update the physical characteristics of the array, such as the dynamic characterisation, and the other one to modify the parameters required for modeling the antenna patterns.

Pressing the ***Antenna Characterization*** label in the *Scenario Definition* box displays the interface shown in Figure V.7. This Antenna Characterization section consists

of two interfaces, called by pushing the ***Array Configuration*** and the ***Pattern and Thermal*** buttons.

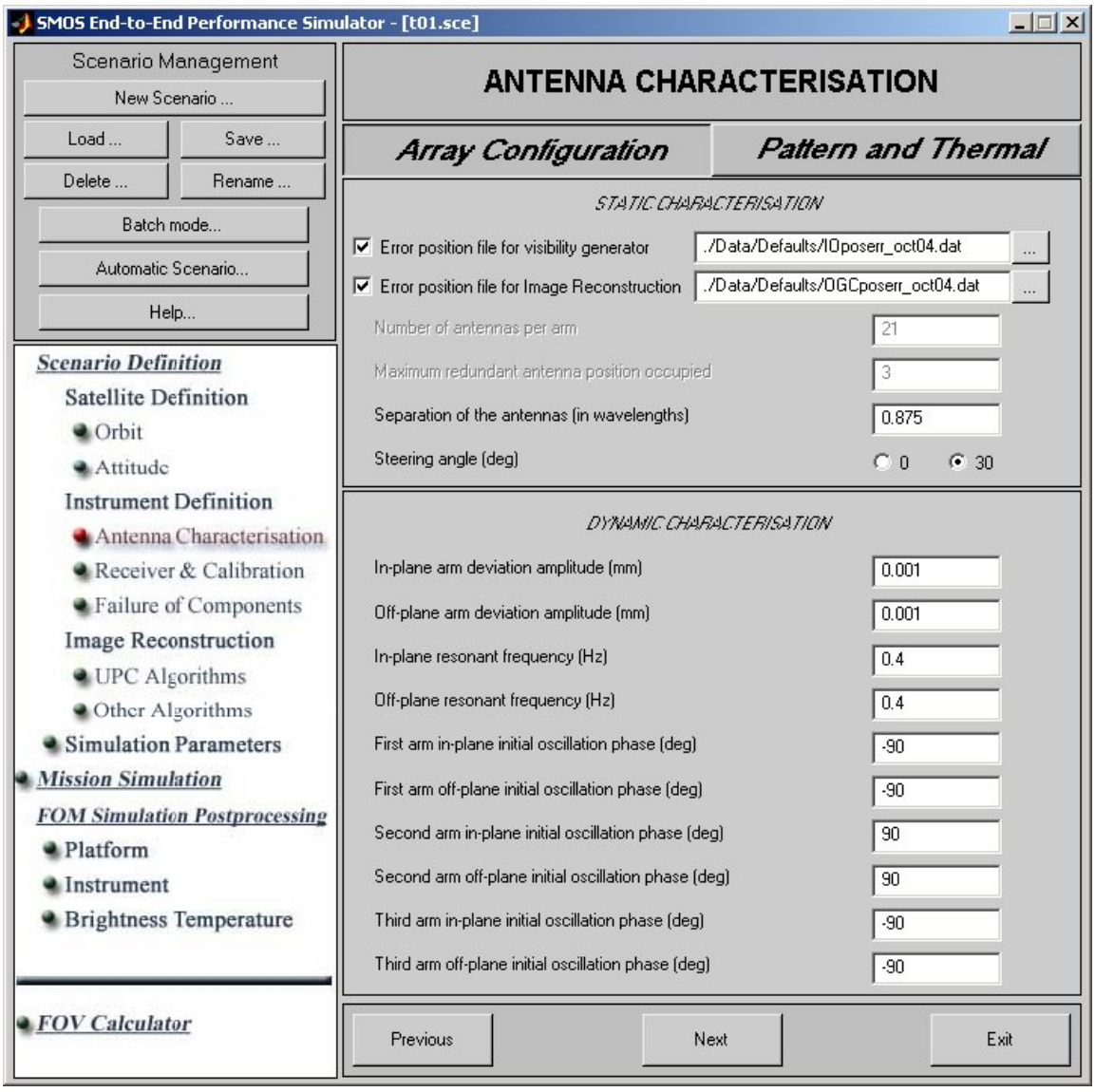


Figure V.7. Antenna Characterization *interface shown by default*

The first interface shown by default is the **Array Configuration**. The interface consists of two parts, for the static and dynamic characterization of the array. The static characterization frame contains the parameters needed to compute the LICEFs' positions in the array, without taking into account the array arms' deformations or oscillations.

The user can select a couple of files containing the static positions of the antenna along the array for observables generator and Image Reconstruction/auxiliary visibilities computation, respectively. To do so, the user must check the small box to the left of the ***Error Position File for visibility generator*** and/or ***Error Position File for Image Reconstruction*** text label, and then press the ***...*** button to open a dialog box through which the file containing the systematic errors (real in orbit and estimated on ground) needed for the static characterization of the array is selected. This action is illustrated in Figure V.8.

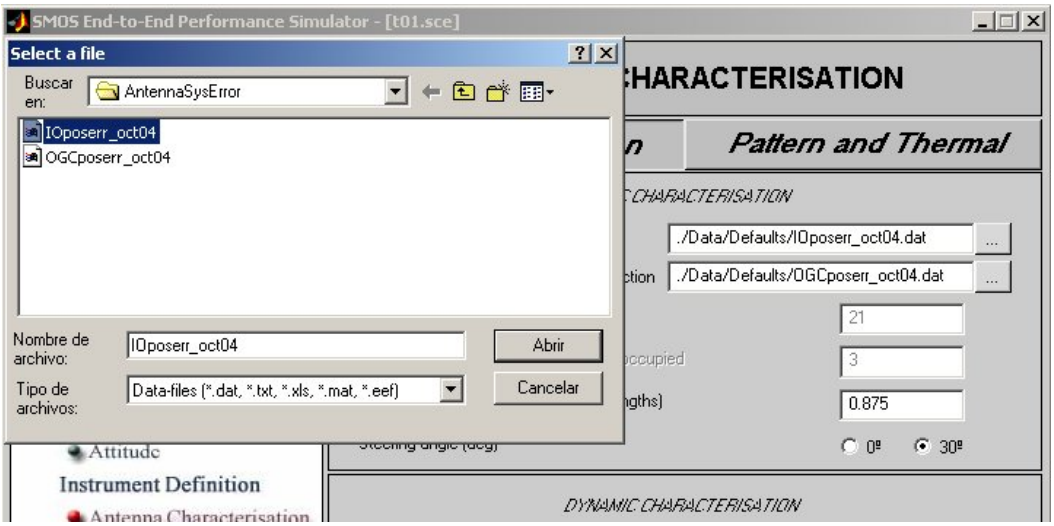


Figure V.8. Dialog box displayed to select the Error Position *files for the array's static characterization*

The other parameters needed for the static characterization of the array are:

- *Number of antennas per arm*: the number of antennas along an array arm, from the normal antenna nearest to the center to the arm's extreme. As in the SMOS Phase C/D present configuration it is set to 21, and cannot be modified.
- *Number of redundant antennas per arm*: the number of redundant antennas in each arm, that is, the antennas from the center of the array to the opposite direction of the normal antennas. As in the SMOS Phase C/D present configuration it is set to 3, and cannot be modified, but it internally assumes that 3 of them (the second of each arm) have failed (see figure V.10).
- *Separation of the antennas*: distance (in wavelengths) between two contiguous antennas. By default is 0.875.

- *Steering angle*: orientation of the array. Its possible values are 0° or 30° (the default value is 30º). The element numbering is shown in fig. V.9 for steering 30º. Elements 2, 26 and 50 are internally assigned as "failing."

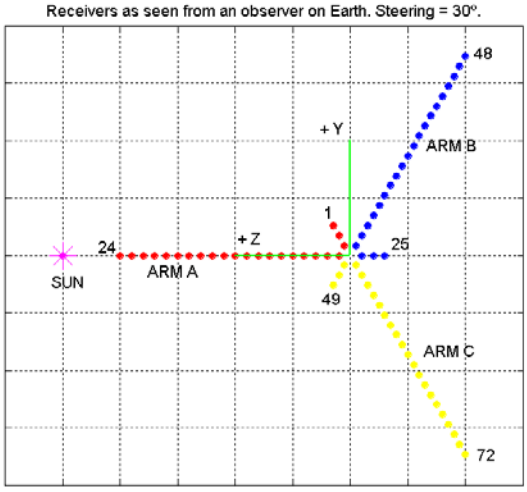


Figure V.9. Receiver and Antenna numbering. To model SMOS PLM (end of Phase B configuration), antennas 2, 26 and 50 are internally selected to "fail." NIR antennas are located in elements' positions 3, 27, and 51, but are internally constituted by two LICEF units located in positions 2-3 for NIR AB, 26-27 for NIR BC and 50-51 for NIR CA.

The orientation of the array determines the shape of the SMOS Field-Of-View (FOV) over the Earth's surface. The image below plots the FOV for a steering of 30° (left hand side plot) and for a steering of 0° (right hand side plot).

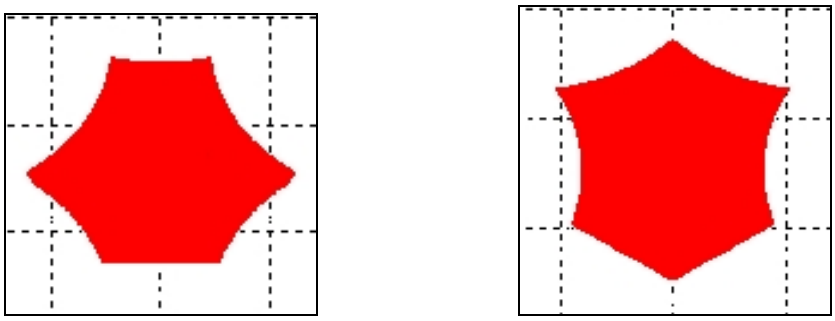


Figure V.10. Field of View (FOV) *with 30º and 0º steering angles, respectively*

For further information on the static characterization of the array, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.12).

The dynamic characterization of the array, that is, taking into account the arms' oscillations and deformations, is done using the parameters in the lower part of the *Array Configuration* interface. These parameters are:

- *In-plane arm deviation amplitude*: amplitude of the in-plane antenna oscillations, in mm.
- *Off-plane arm deviation amplitude*: amplitude of the off-plane antenna oscillations, in mm.
- *In-Plane Resonant Frequency*: in-plane resonant frequency of the antenna oscillation, in Hz.
- *Out-of-Plane Resonant Frequency*: off-plane resonant frequency of the antenna oscillation, in Hz.
- *First arm in-plane initial oscillation phase*: phase of the in-plane oscillation of the first arm at t=0, in sexagesimal degrees.
- *First arm out-of-plane oscillation phase*: phase of the off-plane oscillation of the first arm at t=0, in sexagesimal degrees.
- *Second arm in-plane oscillation phase*: phase of the in-plane oscillation of the second arm at t=0, in sexagesimal degrees.
- *Second arm out-of-plane oscillation phase*: phase of the off-plane oscillation of the second arm at t=0, in sexagesimal degrees.
- *Third arm in-plane oscillation phase*: phase of the in-plane oscillation of the third arm at t=0, in sexagesimal degrees.
- *Third arm out-of-plane oscillation phase*: phase of the off-plane oscillation of the third arm at t=0, in sexagesimal degrees.

For further information on the dynamic characterization of the array, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.13).

Clicking the second upper button of the *Antenna Configuration* section, the ***Pattern and Thermal*** button, displays the interface shown in Figure V.11. This interface is

dedicated to the update of the parameters needed for the definition of the antenna patterns and the thermal characterization of the instrument.

Two alternatives are provided to define the antennas' patterns: using a mathematical function or using a set of stored data files containing the patterns. By default the mathematical function option is selected. Its parameters are:

- *Directivity parameter*: the function characterizing the normalized antenna patterns has the form $\cos^D \theta$, where D is the directivity parameter requested. By default it is 3, which best fits LICEFs antenna patterns.
- *Maximum of the radiation pattern*: the orientation of the maximum of radiation of the antennas, in sexagesimal degrees. By default 0°.
- *Standard deviation of amplitude error*: amplitude ripples standard deviation of the co-polar antenna patterns, in % respect to boresight.
- *Standard deviation of phase error*: phase ripples standard deviation of the co-polar antenna patterns, in sexagesimal degrees.
- *Cross-polar value*: value of the cross-polar term with respect to the co-polar at boresight in decibels.

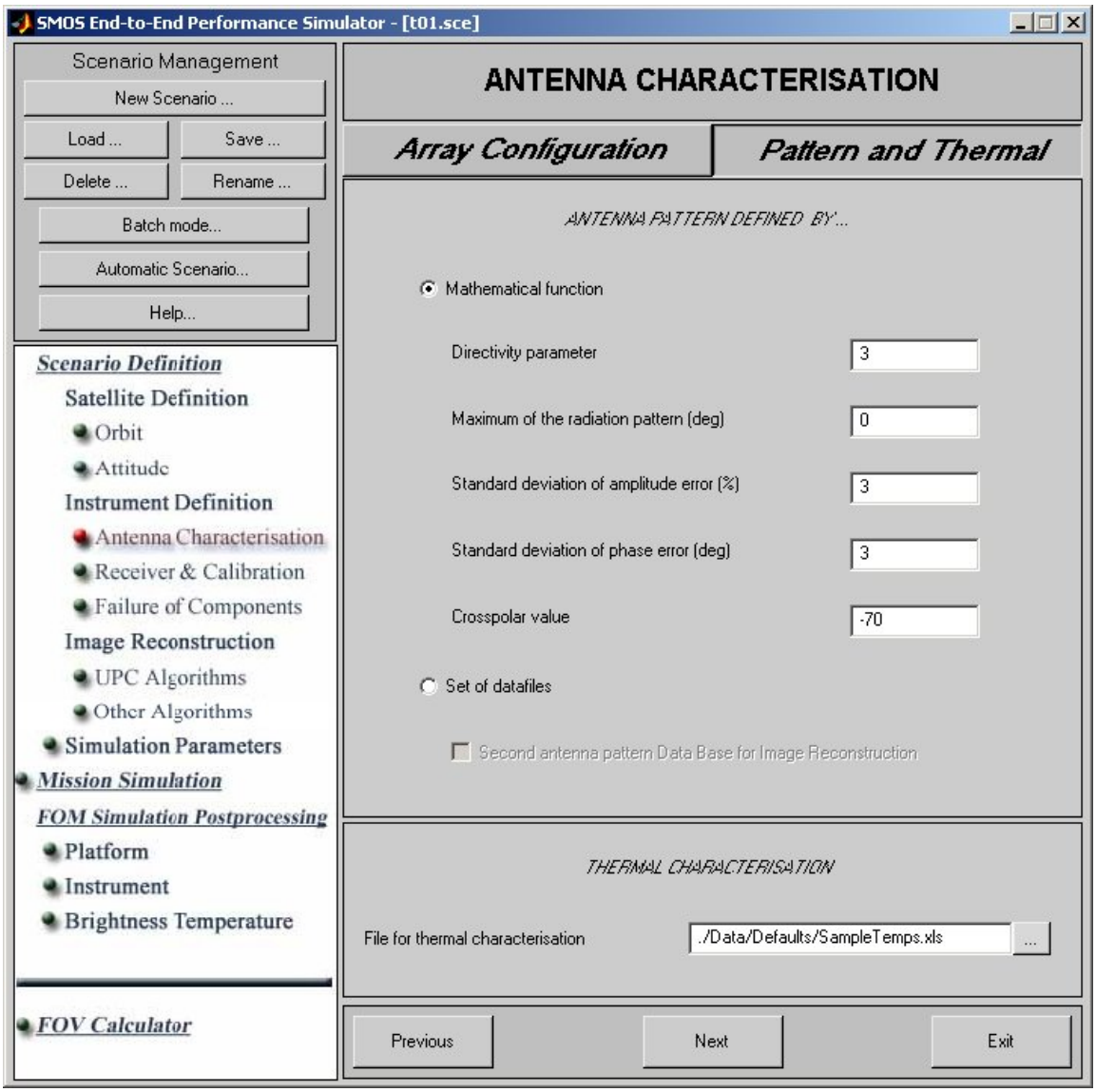


Figure V.11. Pattern and Thermal *interface shown by default*

The other option is to use the data files containing the antenna patterns, to do so the user must click on the ***Set of datafiles*** radio button. For further information on the antenna patterns computation, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.15).

Second antenna pattern Data Base for Image Reconstruction allows the user to use a set of antenna datafiles for image reconstruction that are different from the ones used for visibility generation. The datafiles used for both generation of visibilities and image reconstruction are located in *SEPSroot / Data / Input / DefaultInputFiles / Antennapattern /* folder.

The antenna pattern datafiles have the following naming:

io_patch_cop_x.mat	io_patch_cross_x.mat
io_patch_cop_y.mat	io_patch_cross_y.mat
io_patch_back_cop_x.mat	io_patch_back_cross_x.mat
io_patch_back_cop_y.mat	io_patch_back_cross_y.mat
og_casa_patch_cop_x.mat	og_casa_patch_cross_x.mat
og_casa_patch_cop_y.mat	og_casa_patch_cross_y.mat
og_casa_patch_back_cop_x.mat	og_casa_patch_back_cross_x.mat
og_casa_patch_back_cop_y.mat	og_casa_patch_back_cross_y.mat

where "io" and "og" stand for "in orbit" (used to compute the observables) and "on ground" (used to compute the auxiliary visibilities and to be used in the image reconstruction algorithm), the "cop" and "cross" stand for co- and cross-polar antenna voltage patterns (note: the cross-polar is normalized with respect to the co-polar radiation maximum, as usual), the "x" and "y" stand for the X and Y antenna polarizations, and whenever the term "back" appears in the filename it refers to the antenna back pattern ($90º \leq \theta \leq 180º$, $(\xi,\eta,\omega)|_{back} = (\xi,\eta,-\omega)|_{fore}$). These files contain 181x181 matrices of complex numbers (amplitude and phase) of (ξ,η) points on a uniformly distributed grid from -1 to +1. Values outside the unit circle are set to zero.

At the delivery of SEPSv4.0 the same set of data have been include in both datafiles, but this option allows in the future the use of a set of datafiles with the known (measured on-ground) antenna patterns for image reconstruction and a set of antenna datafiles that takes into account the possible changes in the antenna patterns in orbit.

The lower part of the *Pattern and Thermal* interface is the *Thermal Characterization* section, which consists only in allowing the user to supply a file containing the thermal characterization.

As it can be seen in Figure V.11, the default file provided by the SEPS v4.0 – named as *SampleTemps*, which contains the thermal data for an orbit with maximum temperatures-, either with Excel .xls file format or in .dat ASCII format, is stored in the *SEPSroot / Data / Defaults / folder*. The user can provide a new file by clicking on the right side **...** pushbutton, which opens a dialog box for looking another thermal characterization file, as in the example seen in Figure V.12.

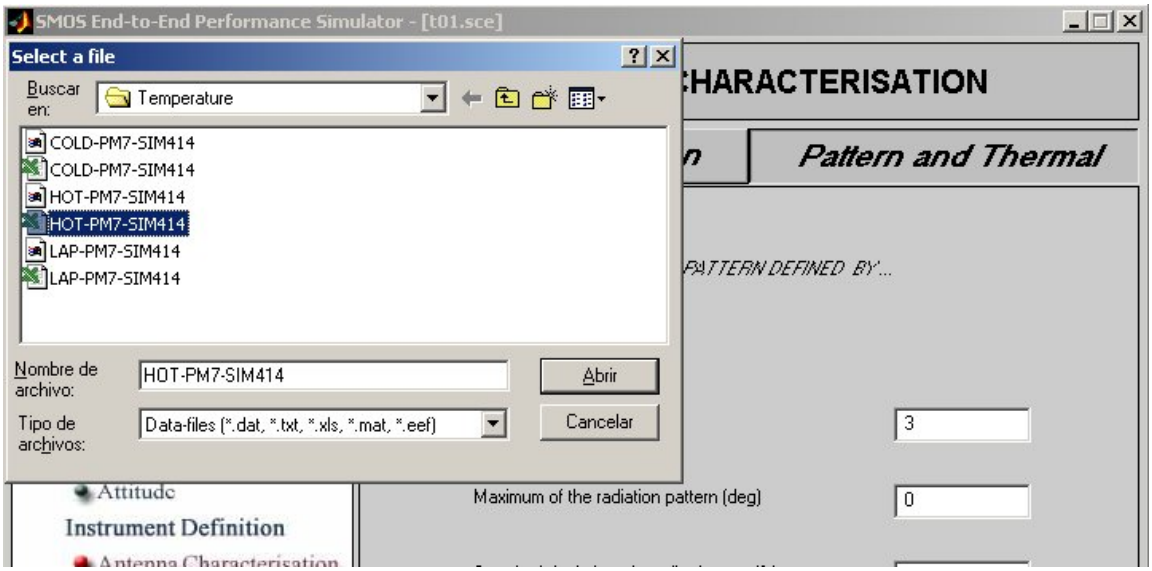


Figure V.12. Dialog box opened to provide a new thermal characterisation file

The format needed for the thermal file is the following:

- The first column contains the time (in seconds) since passage through the ascending node for an entire orbit;
- The rest of columns contain the temperature data of the relevant PLM elements, in degrees Centigrade, for the time along the orbit of the corresponding row:
    - from 2nd to 73rd column, the data included is that of the receivers and the NIR-LICEF boxes, 24 in total for each arm (these include the hub elements that make up each radiometric arm) -note: receivers 2 and 3,

26 and 27, 50 and 51 correspond to the H and V channels of the AB, BC and CA NIRs, respectively-;

- from the 74th to the 145th column, the data included is that of the antenna patches of the elements of each arm. The second element of each arm has dummy data, since the antenna at that location was removed at it was in the best interest to maintain the format of the files;
- from the 146th to the 169th column, the data included is that of the Power Dividers;
- from the 170th to the 179th column, the data included is that of the Noise Sources;
- and from the 180th to the 182nd column, the data included is that of the NIR controller boxes.

For additional information on the contents of the file:

- the actual files, in .xls and .dat format are located in *SEPSroot / Data / Input / UserInputFiles / Temperature /* folder. Inside the files, each column is labelled with the official nomenclature, described in SO-PL-CASA-PLM-0022.
- Look at the AD/DD (SO-TN-GMV-PLM-0003, section 6.14) where a more detailed description is available.

In Figure V.12 it can be seen that three set of data files are provided by default with the SEPS v4.0 software package –each data set either in Excel .xls or in ASCII .dat type-: the one named as *HOT-PM7-SIM414* contains the thermal data for an orbit with maximum temperatures, the one named as *COLD-PM7-SIM414* contains the thermal data with the minimum temperature, and the last one named as *LAP-PM7-SIM414* contains the values for the Low Available Power case.

***NOTE**: the physical temperature file used in the simulation is the one selected in this section. It is the user's responsibility to ensure that the temperature file used is coherent with the orbit/attitude used in the simulation.*

V.2.2. Receiver and Calibration

The definition of the receivers and also the calibration procedures is done using a set of input parameters. These parameters are available to be updated through the interface called when clicking the ***Receiver and Calibration*** label in the *Scenario Definition* box, as shown in Figure V.13.

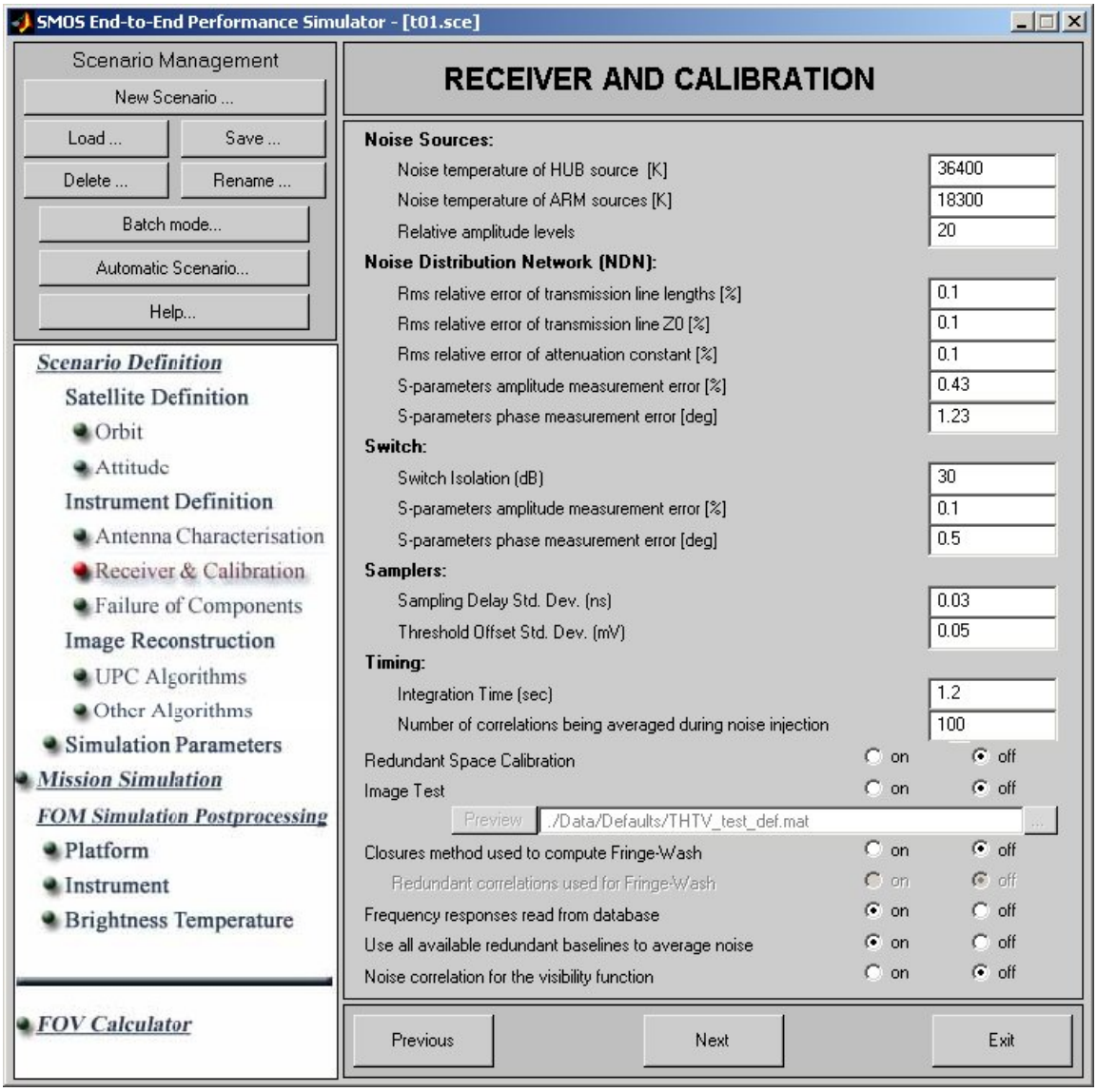


Figure V.13. Default appearance of the Receiver and Calibration *interface*

This interface consists of a set of edit boxes where the user may modify the calibration parameters' values, and a set of mutually exclusive radio buttons for (de)activating calibration and correction procedures. Also, an interactive dialog tool allows selecting the image file to perform the imaging test, as well as performing a preview of it.

Below an explanation of the input parameters (grouped by type) available to be updated through edit boxes is provided.

- Noise source parameters:
  - *Noise temperature of hub source*: the noise temperature being injected by the source to the central hub of antennas for Centralized Noise Injection procedure, and the first step of the Distributed Noise Injection along the arms. Units are Kelvin.
  - *Noise temperature of arms sources*: the noise temperature being injected by the sources to 6-antennas groups along the array's arms, for the performance of the Distributed Noise Injection procedure. Units are Kelvin.
  - *Relative amplitude levels:* of the noise source levels (high and low levels) used in the 4 point calibration.
- Noise distribution networks (NDN) parameters:
  - *Rms relative error of transmission line lengths*: in %.
  - *Rms relative error of transmission line Z0*: the rms error allowed to characteristic impedance of the transmission lines in %.
  - *Rms relative error of attenuation constant*: in %.
  - *S-parameters amplitude measurements error*: in %.
  - *S-parameters phase measurements error*: in deg.
- Switch parameters:
  - *Switch Isolation*: value of the isolation in the polarization switch. Units in dB.
  - *S-parameters amplitude measurements error*: in %.
  - *S-parameters phase measurements error*: in deg.
- Samplers parameters:
  - *Sampling delay standard deviation*: in ns.
  - *Threshold offset standard deviation*: in mV.
- Timing parameters:

- *Integration time*: time interval along which antenna measurements for each polarization are integrated. Units are seconds.
- *Number of correlations being averaged*: the correlations averaged during the calibration process.

The mutually exclusive radio buttons allow the user to (de)select a number of calibration and measurements' enhancement procedures. The following explanation gives the basics of these procedures. Additional information can be found in the AD/DD Document.

- *Redundant space calibration*: if this option is chosen, the *Redundant Space Calibration* method to correct the calibrated visibilities phase is performed during the simulation. It is applied after Noise Injection Calibration to compensate for error in the antenna to polarization switch path.
- *Closures method used to compute the fringe-wash functions*: the user may activate this option for estimating the fringe-wash coefficients using the closures method. Also may activate the *Redundant correlations used for fringe-wash* to enhance the performance of the method using the redundant correlations.
- *Frequency responses read from database*: the receivers' frequency responses can be either generated through a set of parameters, or loaded a set of data files containing the temperature-dependent frequency responses measured for the first 10 LICEF-2 prototypes (repeated sequentially so as to have as many as array elements), activating the *on* radio button.

Note 1 ➔ Shortly after the release of SEPS v.4.0 there will be an update of the DB, in which Flight LICEF data will be included

Note 2 ➔ During the temperature tests performed by CASA and the SMOS project team an anomalous behavior in the quadrature error of LICEFs 3, 13, 23 ... has been detected, which translates in a degradation of the radiometric accuracy away from the reference temperature. In this SEPS release (4.0) the frequency response of receivers 3, 13, 23..., has been replaced by that of receiver 1 in the current database. Future databases associated to SEPS 4.0 will include actual FM frequency responses.

- *Use all available redundant baselines to average noise*: noise can be reduced if the average of the redundant baselines is computed, activating the corresponding *on* radio button.
- *Noise correlation for the visibility function*: add correlated (on) or uncorrelated (off) noise.
- *Image test*: some test images are provided in order to check the correct operation of SEPS. This option forces SEPS to use these images and not the ones generated by the brightness temperature generator. The user must activate the *Image Test on* radio button, and then the file, by examining box below these radio buttons become enabled. The user must press the … button to open a dialog box, as it is seen in Figure V.14, and use it to modify the file for the image test.

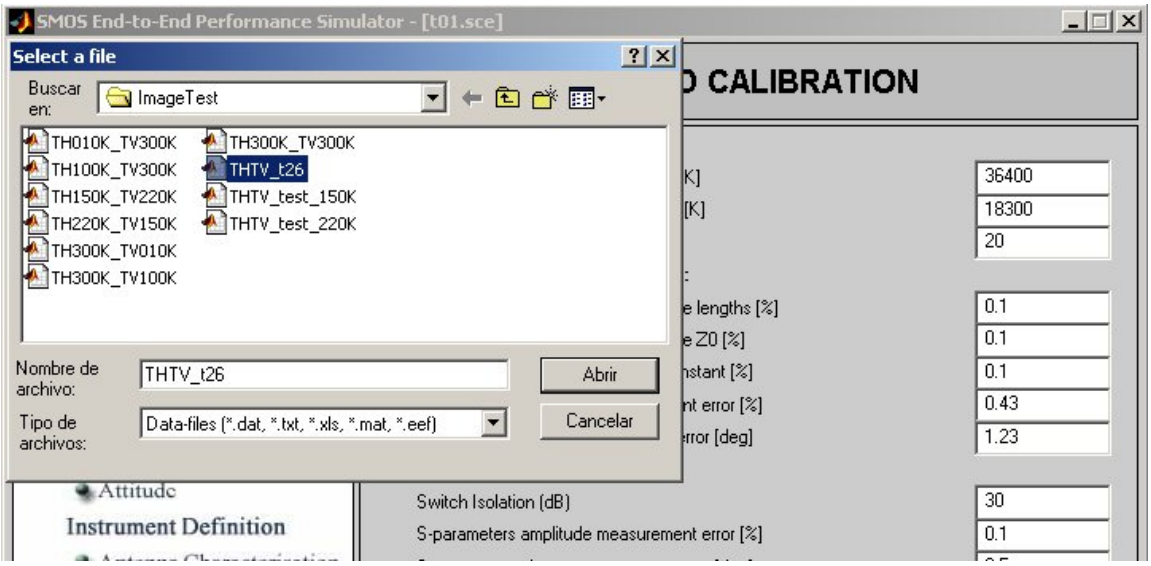


Figure V.14. Dialog box opened to allow the user selecting a new Image Test *file*

The user has the possibility to push the ***Preview*** button to launch a tool to visualize the image test file selected by using the dialog box. An example of the use this previewing tool is shown in Figure V.15. The Vertical and Horizontal polarizations images stored in the .mat file provided by the user as *Image Test* file are shown for each snapshot individually, and a menu option makes the *Preview* tool to interact with the user. Clicking ***Select # Snapshot*** menu several

submenus are opened out. The ***SNAPSHOT #N*** submenu (being N the number corresponding to one of the snapshot stored in the file) leads to displaying the N snapshot image, while clicking ***Select Snapshot …*** submenu displays a small window to manage the snapshots to be displayed (see Figure V.15). This window consists of a list-box, which contains all the snapshots available in the .mat file, and two buttons: ***Apply*** to display the snapshot image selected in the list-box, and ***Close*** to close the window.

On the other hand, clicking the ***Quit*** menu forces to exit the *Preview* tool.

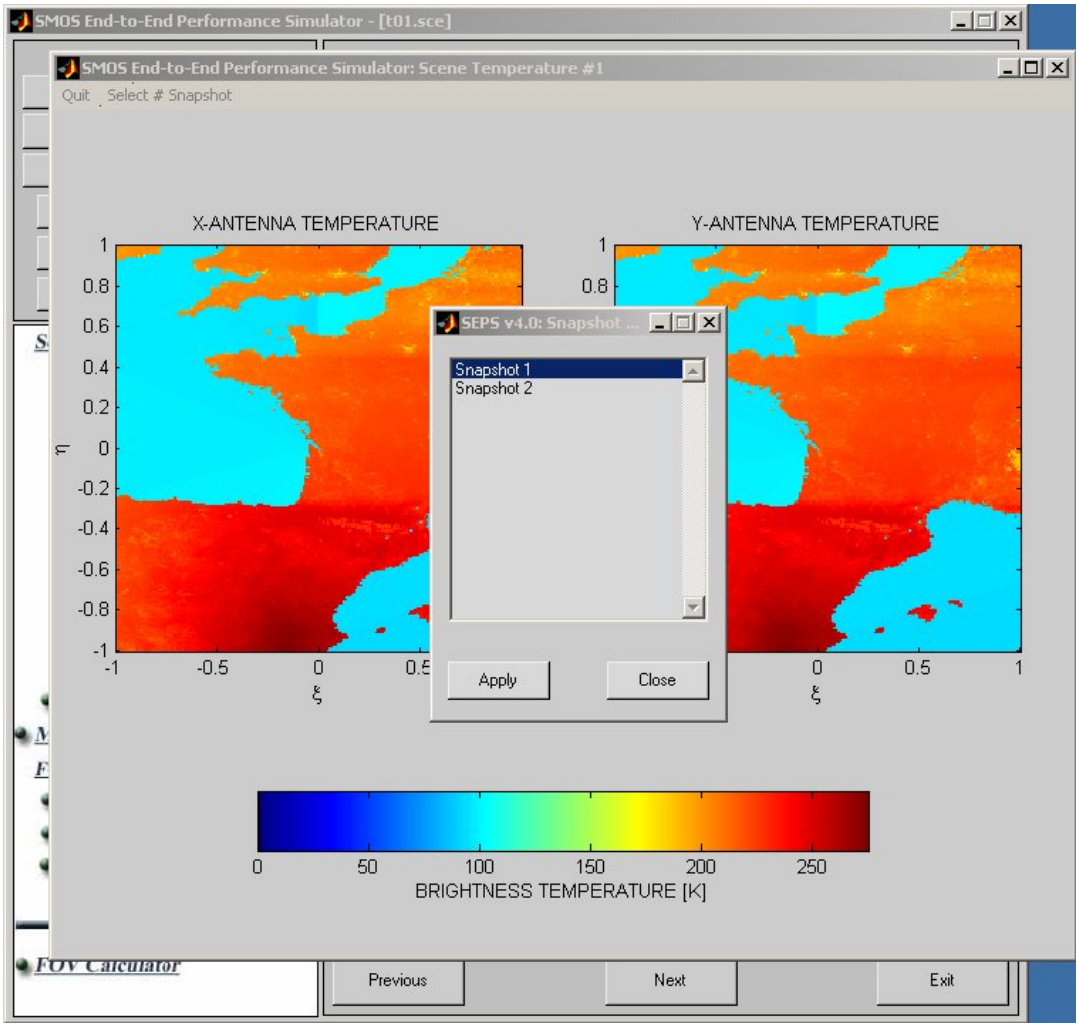


Figure V.15. Preview *of the Vertical and Horizontal polarization brightness temperature images selected by the user for the* Image Test.

The user may notice that in the Image Test mode, the usage of the Cartographic Maps module is not allowed. This is due to the fact that all the images in the image test file are located in the same geographic coordinates, and therefore their geographic projection would be overlapped.

For further information on the calibration procedures, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, sections 6.23 to 6.26).

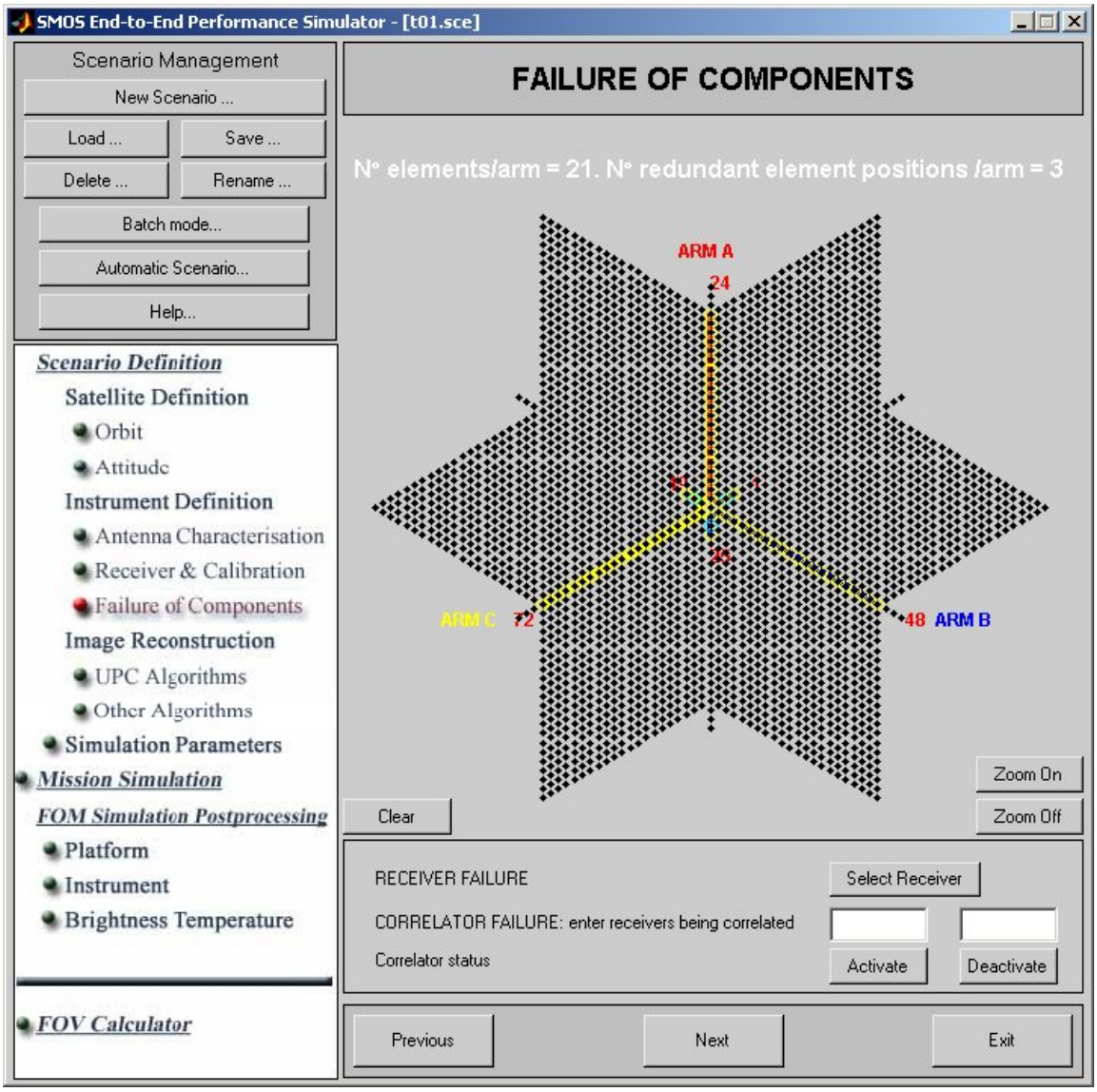


Figure V.16. Default appearance of the Failure of Components *interface when steering angle is 30.*

V.2.3. Failure of Components

SEPS allows the user to make certain array's elements fail in order to evaluate its performance under malfunction conditions. Clicking the ***Failure of Components*** label in the *Instrument Definition* section of the *Scenario Definition* box, the interface shown in Figure V.16 is presented to the user.

The array elements orientation is shown by the numbers in red color, which shows the numbering of the antennas at the edges of each array's arm. The receivers are shown as yellow circles along the array arms, while the correlators are symbolized by the black points located at the (u,v) points corresponding to the visibilities obtained from those correlators.

A correlator computes the correlation between the output signals from a pair of receivers, therefore obtaining the visibility function value for the corresponding (u,v) point. If a correlator is failing, its output signal is not available as a valid measurement. But that measurement can also be unavailable if one or two of the receivers' output signals being correlated are not available. When a receiver is forced to fail all its dependant correlators – those using the output signal of that receiver as one of the two inputs to be correlated- cannot provide a valid output. Then, deactivating a receiver makes all its dependant correlators to be unavailable.

To activate or deactivate a receiver the user may press the ***Select Receiver*** button, and then click the left mouse button on the yellow circle that represents the selected receiver, which becomes deactivated if it was previously operative, and becomes activated if the receiver was previously failing. In the case the receiver becomes deactivated, the circle identifying it in the interface changes from yellow to cyan color, and the visibilities measured from the correlation of the failing receiver's signals and other receivers change from black to green color to indicate that those visibilities are not available anymore. In the case the receiver gets activated, the circle representing the receiver turns to yellow and its dependant visibilities turn to black.

Figure V.17 shows the result of deactivating two receivers. It can be observed that the (u,v) points where there are not redundant visibilities become in green, to represent that no measurements can be obtained in that position for the visibility function. The positions where redundant measurements exist –as in the ones nearest to the arms-, remain in black.

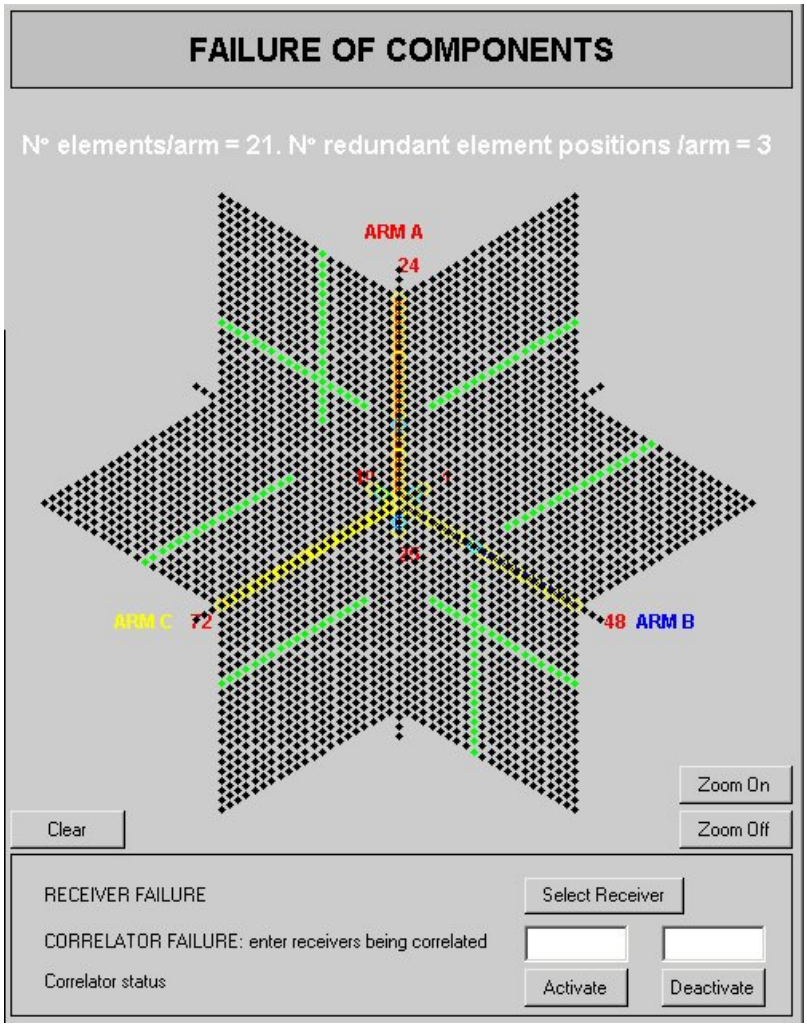


Figure V.17. Failure of Components *interface after deactivating two receivers. Only the (u,v) points where no redundant visibilities exist become invisible.*

The user has to take into account that instrument's calibration may be impossible for some combinations of failing receivers, making the simulation to abort. The elements in the hub and the arms of the array are divided in sections (5 in the hub, and 6 in each arm

section). Noise sources inject correlated noise to two adjacent sections, and phase tracking between adjacent them is achieved by switching the noise sources (odd → even) that are injecting correlated noise. If calibration fails in one section, outer sections in the same arm could not be calibrated, and instrument's performance is degraded. If calibration fails in the hub, the phase tracking between the arms in the array cannot be accomplished, the simulation is terminated, and the message *"SIMULATION ABORTED. Calibration is not possible, please check the receivers' failures combination."* is displayed in the MATLAB Command Window.

Also, the elements that obtain the visibility measurements from the correlation of two receivers' signals, the correlators, can be deactivated. To activate or deactivate a single correlator, the user may supply the receivers' identifying numbers in the *CORRELATOR FAILURE* edit boxes: the selected correlator is the one correlating the signals of the supplied receivers. The user may press then the ***Activate*** or ***Deactivate*** buttons to make the correlator respectively operative or fail. Figure V.18 shows the resulting interface after deactivating a correlator –in this case, correlating receivers 11 and 39 outputs-. White circles have been added in this figure around the green points controlled by the correlator to emphasize the effects of this operation.

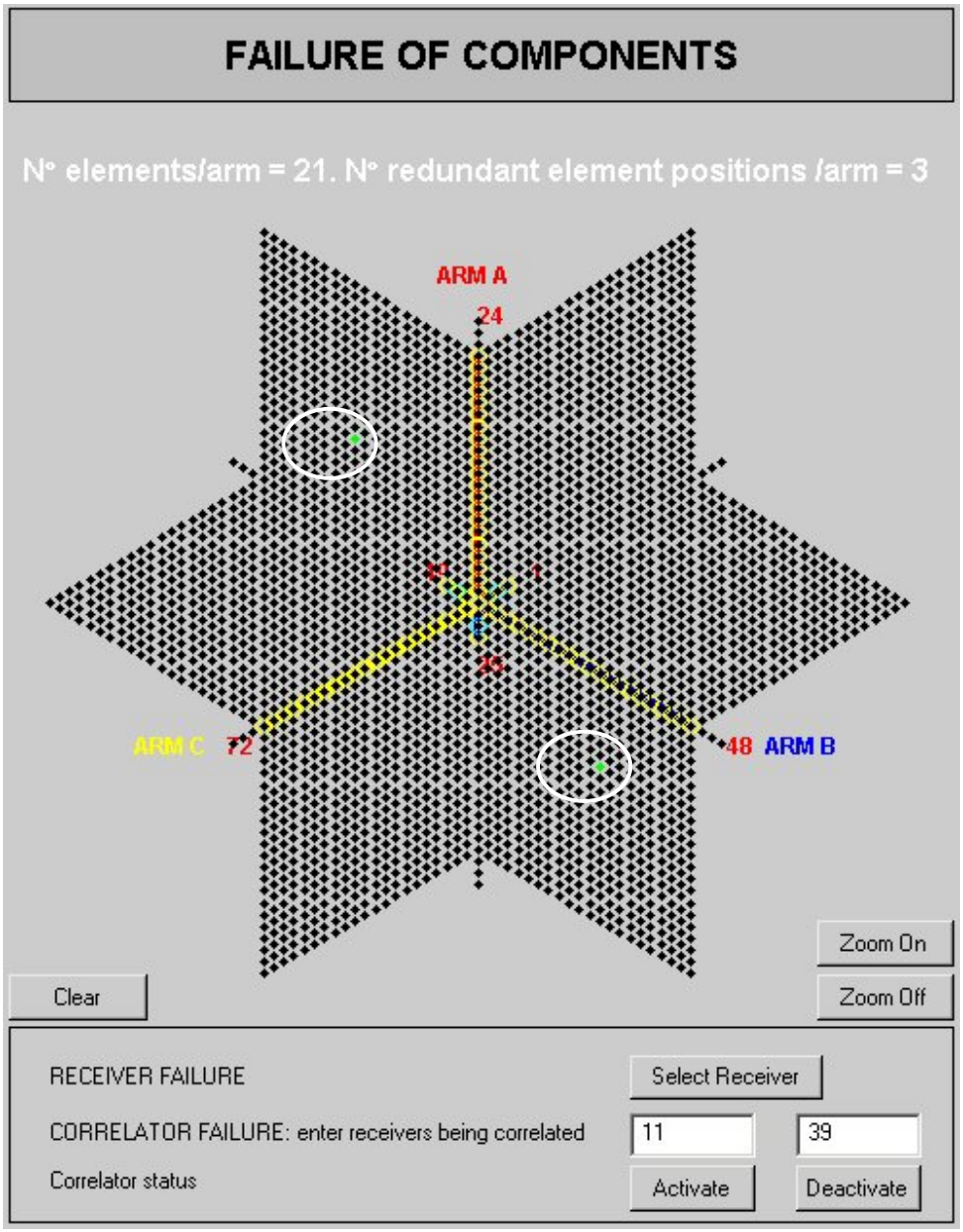


Figure V.18. Failure of Components *interface after deactivating a correlator. White circles are placed around the green points, representing visibilities that are not measured because of this failure.*

A correlator controls two visibilities, a visibility determined by the receivers' outputs being correlated, and also its hermitic one. Then, in the case the correlator becomes deactivated, the black points identifying the two visibilities controlled by that correlator

change to green points if there is not a redundant visibility in that position –that is, other pairs of receivers result in a visibility at the same position-, but remain unchanged if there exists at least one. In the case the correlator becomes activated, both black points identifying the two visibilities controlled by the correlator maintain the black color –although redundant visibilities are previously present, and therefore the black points already exists-.

Also a *Zoom* tool is present in order to view the plot closer, and hence to select the activating/deactivating devices in a more comfortable way. For this purpose, the user may press the ***Zoom On*** button and click the left mouse button over the plot area to be enlarged. A more precise zooming can be performed if the user holds the left mouse button clicked and describes a rectangle enclosing the area to be enlarged. Figure V.19 shows two plots illustrating both the action of selecting an area to be enlarged and the results of this action. The user may press the ***Zoom Off*** button show back the complete array.

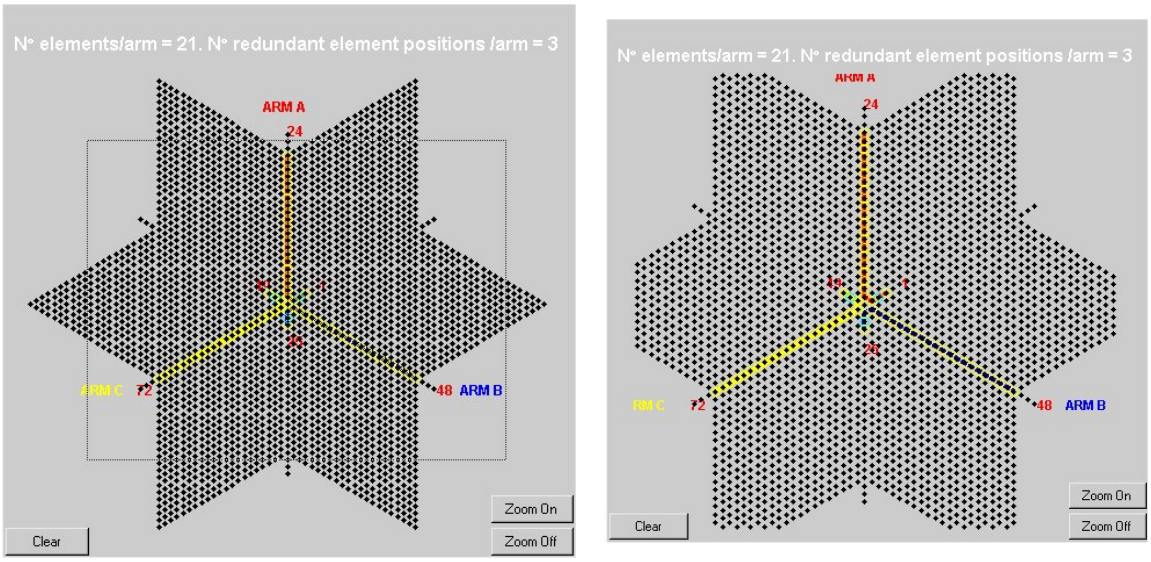


Figure V.19. Left: selection of a rectangle enclosing the area to be zoomed, after clicking the Zoom On button. Right: resulting appearance after zooming.

Finally the ***Clear*** button forces all the failing devices, either receivers or correlators, to be operative again, in order to make the system work in proper conditions.

VERY IMPORTANT ➔ *the **Clear** button does not activate the redundant receivers numbered as 2, 26 and 50, because these receivers are not present in the current SMOS PLM (MIRAS instrument) configuration. However, the user has the possibility to activate these receivers again, in order to evaluate the impact on system's performances of keeping them in the array's configuration.*

For further information on the failure of components, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.22).

V.3. Image Reconstruction

In this section the input parameters needed to define the image reconstruction algorithm to be applied during the simulation are described. Three image reconstruction algorithms are available through SEPS v4.0, supplied by UPC (*Technical University of Catalonia*). For further information on these topics, please refer to the AD/DD document.

***Note**: Image reconstruction algorithms are devised to operate in nominal local normal + yaw steering mode with arbitrary tilt angle. Inaccurate results will be obtained in other situations. In particular, if the Earth is not in the instrument's alias-free FOV, but it is in the FOV (unit circle), bias can be significant*

V.3.1. UPC Algorithms

The *Universitat Politècnica de Catalunya* (*Technical University of Catalonia*), UPC supplies two image reconstruction algorithms to this fourth release of SEPS: the Extended-Clean method –already present in the first release of SEPS- and the G-Matrix method –already present in the second release of SEPS-. To show the input parameters interface where the definition of both algorithms is done, the user may click the left mouse button on the ***UPC Algorithms*** label, and then the interface shown in Figure V.20 is presented.

The *Extended-Clean* is an iterative differential algorithm that uses the measured visibilities to retrieve the apparent brightness temperature image at the antennas frame, applying corrections referent to the measured fringe-wash functions, the antenna patterns,

the obliquity factor, etc… (please refer to the AD/DD document, ref. SO-TN-GMV-PLM-0003, sections 6.31 to 6.38, for a complete description of this algorithm).

The *G-Matrix* algorithm, as its name indicates, builds a matrix where each row corresponds to a measured visibility, and each column to a point of the cosines domain. In fact it is an application of the visibility function formula in algebraic form. Then, the Conjugate Gradient Square method is applied to solve the system of equations (please refer to the AD/DD document, ref. SO-TN-GMV-PLM-0003, sections 6.39 to 6.44, for a complete description of this algorithm).

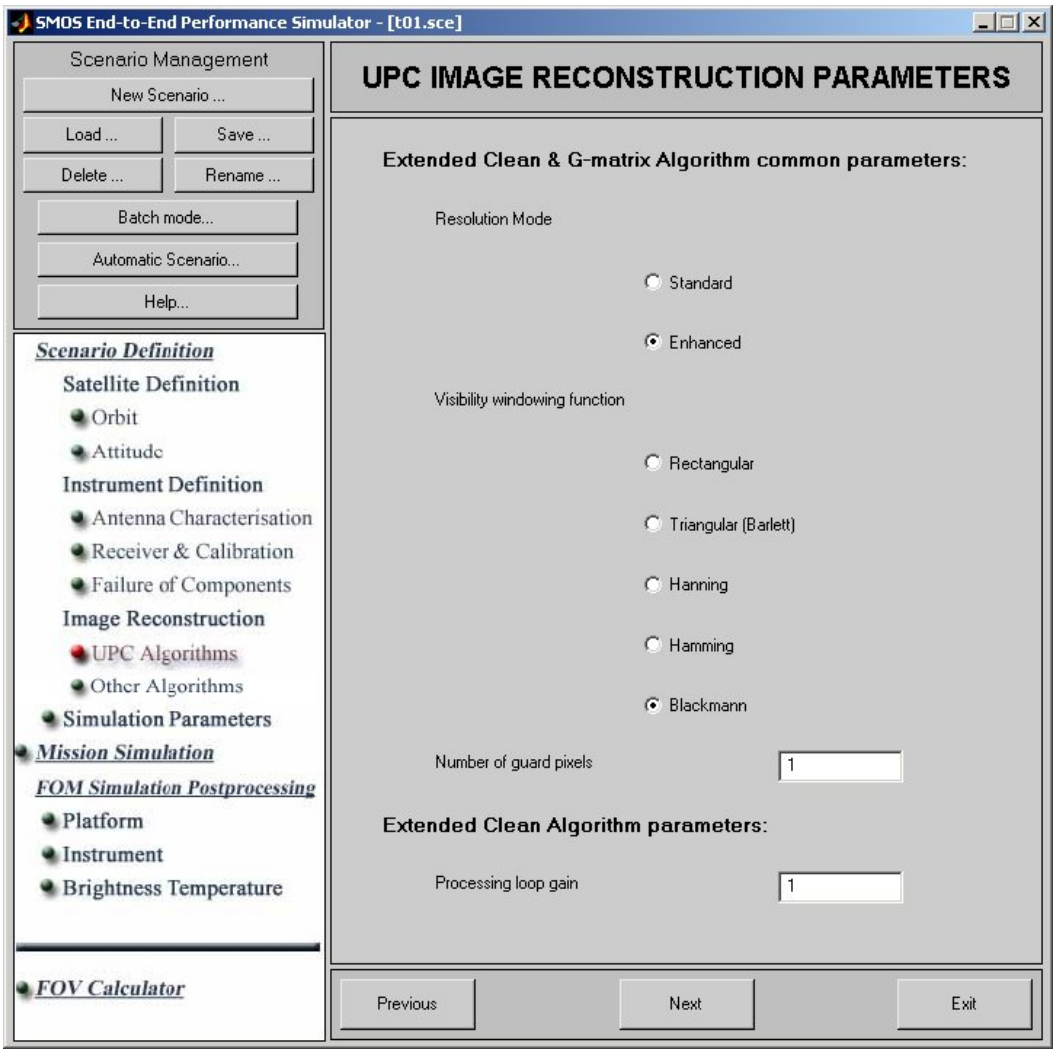


Figure V.20. UPC Algorithms *interface.*

The definition of the UPC algorithm is done through a set of mutually exclusive radio buttons to select alternative options, as well as a couple of edit boxes to enter the suitable values for specific input parameters. These input options and parameters are described below.

- *Resolution mode*: this parameter allows the user to define the standard or the enhanced mode, achieved by using the visibilities inside an hexagon inside the full (u,v) star-shaped coverage, selecting the *Standard* mode (lower angular resolution, lower sidelobes), or all the available (u,v) points (6-point star-shaped), if the *Enhanced* mode (higher angular resolution, higher sidelobes) is chosen.
- *Visibility windowing function*: this function is used to taper the visibility samples. The default window is the Blackmann window.
- *Number of guard pixels*: establishes the number of pixels in the earth alias border that are not processed (minimize Gibbs effects). The user may enter the value in the corresponding edit box.
- *Processing loop gain*: this factor is used in the Extended-Clean algorithm to guarantee convergence in extreme cases. The algorithm rarely diverges, mostly under large antenna amplitude and phase errors. A factor <1 usually prevents divergence.

For further information on the UPC image reconstruction algorithms, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, sections 6.31 to 6.44).

V.3.2. Other Algorithms

This option is a placeholder to include future image reconstruction algorithms.

Figure V.21. Other Algorithms *placeholder.*

V.4. Simulation Parameters

The definition of the simulation parameters is performed through two interfaces, one to update the generic simulation parameters, such as Sun and Moon effects, and the other one to define the timeline to be simulated.

Pressing the ***Simulation Parameters*** label in the *Scenario Definition* box displays the interface shown in Figure V.22. This Simulation Parameters section consists of two interfaces, called by pushing the ***Simulation Parameters*** and the ***Modes And Timeline*** buttons.

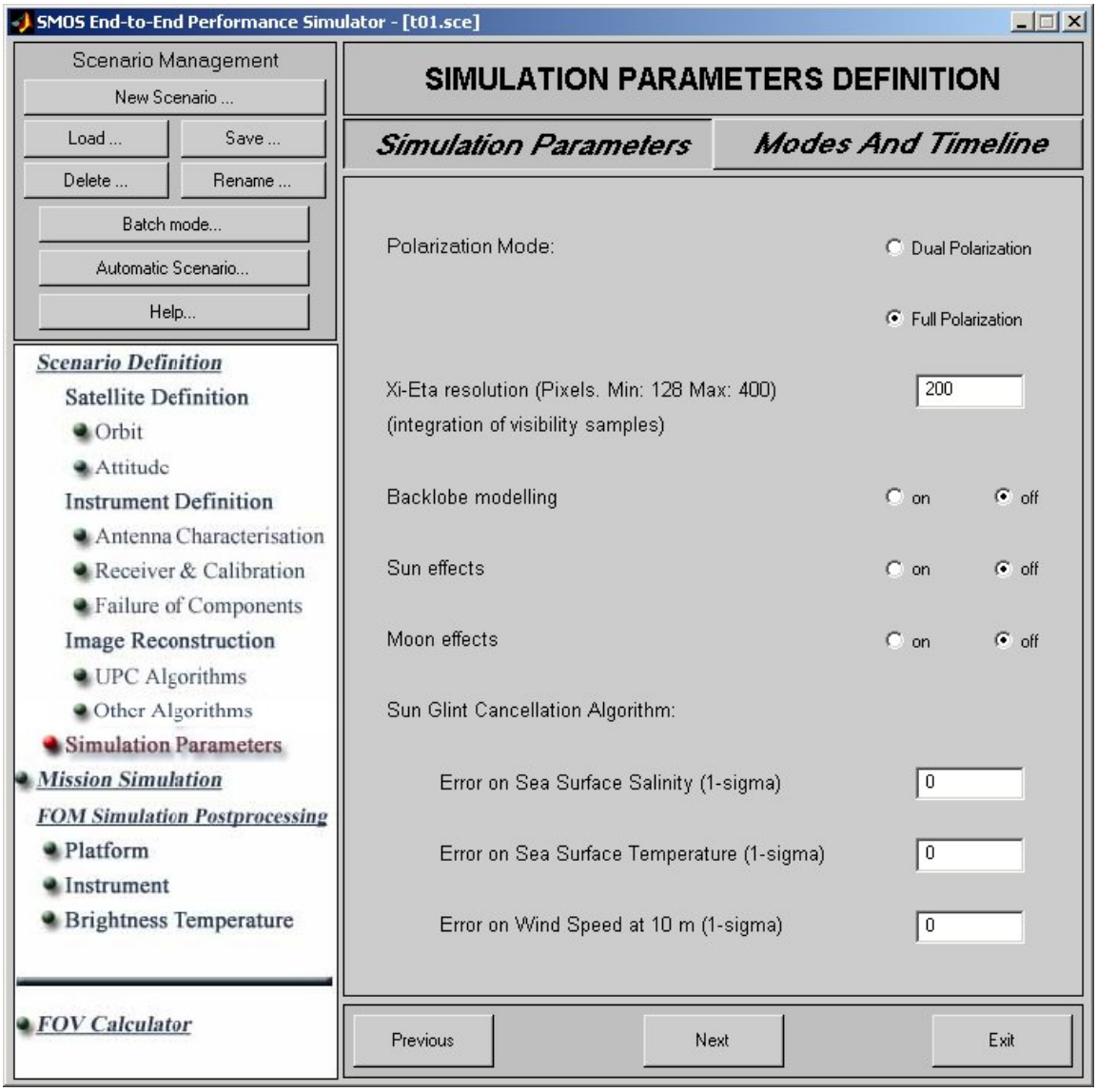


Figure V.22. Simulation Parameters *interface.*

The first interface shown by default is the ***Simulation Parameters***. The interface contains the generic simulation parameters needed to execute the simulation. The input parameters in this interface are described below.

- *Polarization Mode*: two alternatives are available, the Dual Polarization mode (the X and Y polarizations are computed in the antenna reference frame), and the Full Polarization mode (also the Cross-polar polarization is computed, enhancing the temperature image retrieving on Earth, among other benefits).
- *ξ-η resolution*: number of points of the director cosines domain image used to compute the visibility samples double integrals. The larger the number of points, the better accuracy. Note that simulation time increases with the square of the ξ-η resolution. A ξ-η resolution of 128 provides good results. A ξ-η resolution of 64 provides acceptable, but not as accurate results. The range of values is determined from the number of antennas per arm. The ξ-η resolution must be an even number –if an odd number is supplied through the edit box, the interface forces it to be the upper even number.
- *Backlobe modelling*: turning this option to "on" forces the simulator to compute the backlobe antenna effects.
- *Sun effects*: turning this option to "on" forces the simulator to take the Sun effects –such as the Sun reflection on the Earth's surface, or the direct view of the Sun in the antenna's diagram-. Turning it to "off" then forces the simulator to override the Sun effects.
- *Moon effects*: turning this option to "on" forces the simulator to take the Moon effects –such as the Moon reflection on the Earth's surface, or the direct view of the Moon in the antenna's diagram-. Turning it to "off" then forces the simulator to override the Moon effects.
- *Sun Glint parameters*: the user may update the driving parameters of the sun glint model (error on sea surface salinity, error on sea surface temperature and error on wind speed at 10 meters of altitude, all of them expressed as 1-sigma error).

Clicking the second upper button of the *Simulation Parameters* section, the ***Modes And Timeline*** button, displays the interface shown in Figure V.23. This interface is dedicated to define the timeline that will control the execution of the simulation.

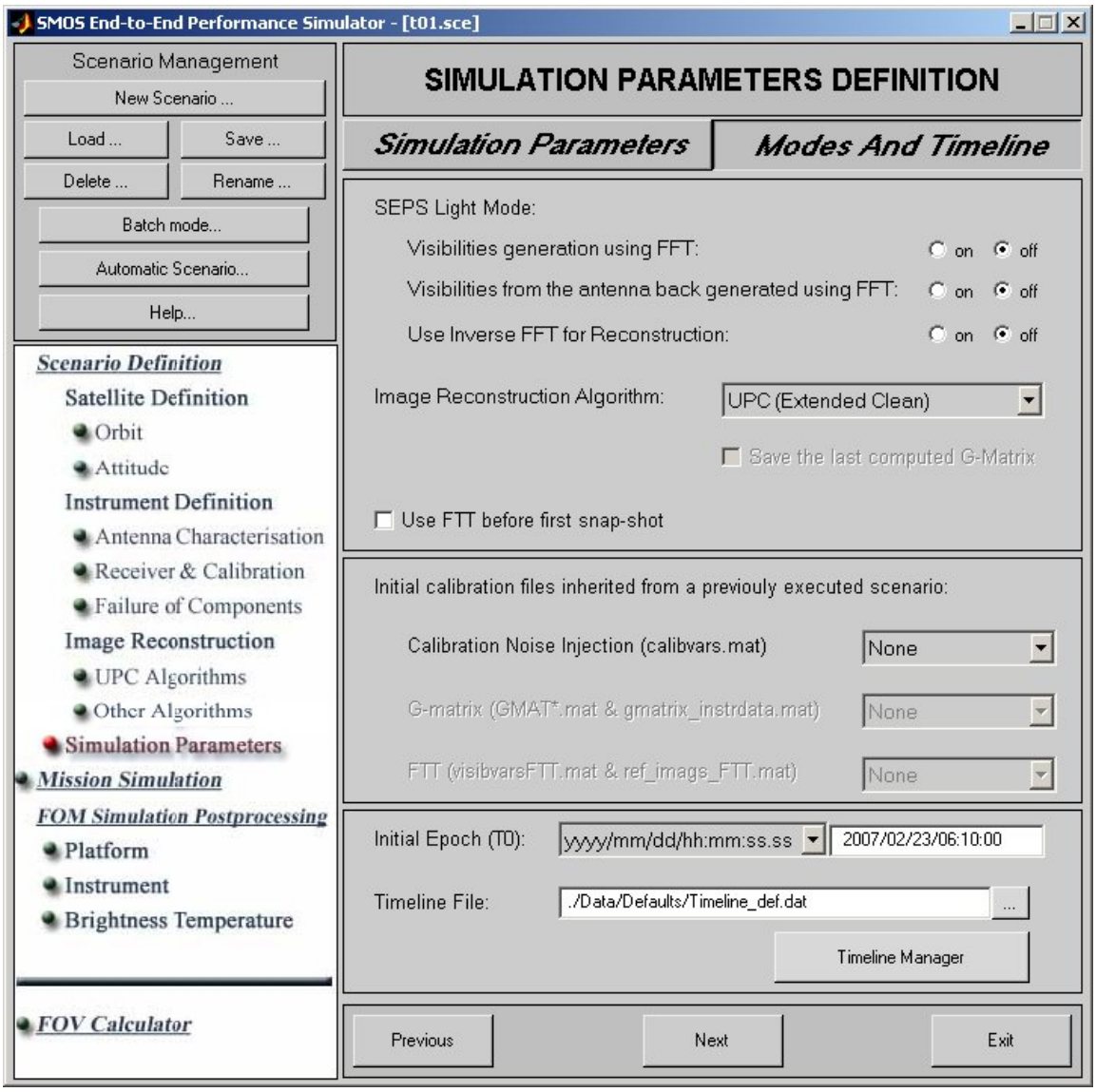


Figure V.23. Timeline *interface.*

In the first panel the user may select the Image Reconstruction method to be used in the simulation. Two options are envisioned: use of the SEPS Light mode or one of the Image Reconstruction methods included in SEPS.

Six radio-buttons allows the user activate or deactivate the **SEPS Light Mode**. The SEPS Light mode allows the user to perform simulations using Fast Fourier Transforms (FFT) to generate the visibility samples, and inverse FFTs to reconstruct the brightness temperature scene, hence improving dramatically the computational speed –at the cost of losing accuracy and capability of modeling several system's imperfections (antenna pattern differences, shape of the fringe-washing function, and antenna positioning errors)-.

If the user sets the ***Visibilities generation using FFT*** to *on*, Fast Fourier Transform is used to generate all visibility samples. The gain in computational speed is obvious, however the resulting visibilities will not be realistic. Selecting the *off* radio button in this option makes the Observables Generator module to compute more realistic visibilities, taking into account the difference between antenna patterns, the shape of the fringe-wash function, among others, at the cost of dramatically decreasing the computational speed. Besides, the user may set the ***Visibilities from the antenna back generated using FFT*** to active the computation of the back-lobe antenna using a Fast Fourier Transform (FFT).

In the other hand, the ***Use Inverse FFT for Reconstruction*** option allows the user to select whether to use an Inverse FFT (selecting the *On* radio button) or a realistic image reconstruction algorithm (*Off* radio button) to retrieve the brightness temperature scene. Again, using the Inverse FFT makes the calculations to be much faster, although losing the capability of taking into account the realistic modeling of system's imperfections. On the other hand, setting this option to *Off* forces to use the selected image reconstruction algorithm selected in the ***Image Reconstruction Algorithm*** popup, and a sensible decrease in computational speed can be easily expected.

Setting the ***Visibilities generated using FFT*** to *on* makes the ***Use Inverse FFT for Reconstruction*** to be *On*. It is forced to be this way because it would be a nonsense to use a set of non-realistic visibility samples to retrieve a brightness temperature scene using an image reconstruction algorithm that takes the modeling of realistic system's imperfections into account.

If the ***Use of inverse FFT for Reconstruction*** is set to *off*, the user may then choose between one of the UPC Image Reconstruction Algorithms –either the ***Extended-Clean*** or the ***UPC CGS (Conjugate Gradients Squared) G-Matrix*** algorithm-

In case of the *UPC CGS G-Matrix* algorithm had been selected, the user may check the ***Save the last computed G-matrix*** check-box in order to save to Matlab files the last G-matrices computed during the simulation (normally, these files –larger than 100 Mb- are removed from hard disk at the end of the simulation) to be re-used in future simulations.

The last parameter of this first panel that the user may update is the ***Use FTT before first snap-shot*** [Martin-Neira, 2004; Suess, 2005]. When the user checks the *Use FTT before first snap-shot* check box the instrument is first pointed to zenith during one snap-shot so as to measure the so-called "Flat Target." From these measurements, with a proper scaling factor, and the computation of the "ideal" sky visibilities, a Fast Fourier Transform can be applied to the subsequent measurements to try to compensate for antenna pattern errors. See appendix D for advanced topics on the FTT.

In the second panel, the user may select initial calibration files inherited from a previously executed scenario. Three types of files can be inherited from previously executed scenarios:

- *Calibration Noise Injection* files: calibvars.mat.
- *G-matrix* files: GMAT*.mat & gmatrix_instrdata.mat.
- *Flat Target Transformation* initialization file: visibvarsFTT.mat

For each of these types of files there is a popup menu to choose the scenario from which obtaining the inherited files. SEPS searches in the SEPSroot \ Data \ Output \ Bin folder all the scenarios which contains the available files to be inherited, and fills with them the popup menus. The default option is *None*, which means that no inherited files will be used in the simulation.

Note*: Only the scenarios executed with the options "Save the last computed G-matrix" and "full-polarimetric" can be re-used as inherited scenarios for G-matrix files.*

The popup menu corresponding to *G-matrix* files is enabled when the CGS G-Matrix algorithm is selected in the *Image Reconstruction Algorithm* popup menu, in other case it is disabled. At the same way, the *Flat Target Transformation* popup menu remains disable while the *Use of Flat Target Transformation* check-box is unchecked.

Finally the last panel allows the user to update the timeline parameters:

- *Initial Epoch*: the epoch at which the timeline file is referred. The user may choose the format of the initial epoch, and then update its value. Allowed formats are: calendar (yyyy/mm/dd/hh:mm:ss); Modified Julian Date of 2000; and initial orbit and time since ANX (this option is only available whether the orbit propagation is initialized from OSF file).
- *Timeline File*: this file is an ASCII file containing the event-record to be simulated. An event-record is a set of events executed periodically from a start time and separated a same time step. Each row in the file defines an event-record, the format of these rows is:
  - 1st column: "use" flag to indicate if the event record is to be computed (1 Yes, 0 No).
  - 2nd column: start time since Initial Epoch for the event-record, in seconds.
  - 3rd column: time step between events of the same event-record, in seconds.
  - 4th column: number of events of the event-record.
  - 5th column: "Snapshot" computation for each event of the event-record (1 Yes, 0 No).
  - 6th column: "Calibration Noise Injection" computation for each event of the event-record (1 Yes, 0 No).
  - 7th column: "G-Matrix" computation for each event of the event-record (1 Yes, 0 No).
  - 8th column: "Flat Target Transformation" reference data computation for each event of event-record (1 Yes, 0 No).

  Every event-record is independent of the rest of event-records, therefore they can be chronologically mixed up inside the timeline file (at the beginning of the simulation, SEPS obtains all the events from the selected event-records and sorts them chronologically to be simulated).

  All the event-records are referred to the *Initial Epoch* defined in the GUI.

The events, which have the G-Matrix computation activated (seventh column), only will be executed if the *G-Matrix CGS Image Reconstruction* is selected as *Image Reconstruction Algorithm*, in other case this flag is ignored.

The events, which have the Flat Target Transformation reference data computation activated (eighth column), only will be executed if the *Use of Flat Target Transformation* is checked (for these events the instrument boresight will be automatically forced to point to the zenith direction), in other case this flag is ignored.

If in the first chronological event it is not activated the calibration flags (sixth, seventh and eighth columns) and the calibration files are not inherited from a previously executed scenario, then SEPS automatically forces to perform a calibration task at this event time (in order to obtain the calibration files) corresponding to the flag activated in the GUI (G-Matrix, FTT...).

The ***Timeline Manager*** button allows the user to see, generate and edit a timeline file. Clicking this button display a window shown in Figure V.23 to manage the timeline files.

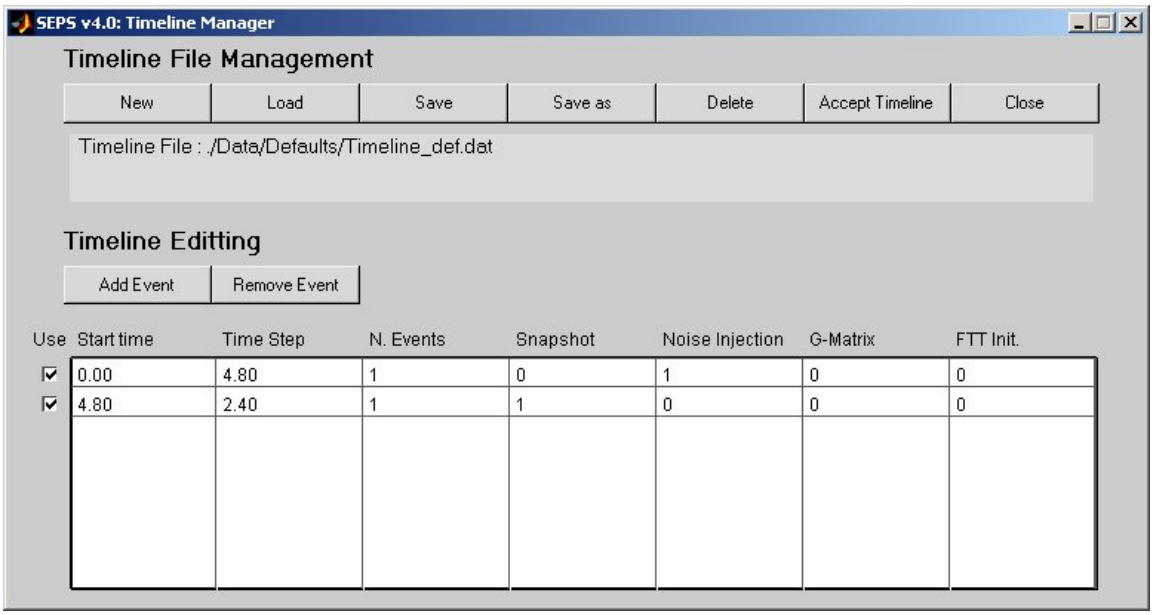


*Figure V.24.*Timeline *Manager Window.*

The user has the use of seven buttons for timeline file management and an editable and scrolling table for managing the timeline files and the event-records. The editable table provides to the user the capability to define and edit event-records, this table contains seven columns in the same way as the ASCII timeline file (see above). Two buttons (on the upper-left corner) allow user to add one event-record (***Add Event*** button) or remove the selected event-record (***Remove Event*** button). When user clicks on a cell on the table, the cell changes colour and is selected. When user double-clicks on a cell the editing cursor appears and user can edit the cell contents. To finish editing, the user may press Enter key or click out of the selected cells. The user may check an event-record clicking on its check box, when the check-box is unchecked the event-record is saved in the file but it is not executed during the simulation.

The user may press the ***New*** button to generate a new timeline file (the default file contains an unique event-record with only one event and all the flags activated). A dialog box is opened where the user may select a file and press the dialog box's ***Open*** button to continue performing the timeline file generation action, or the ***Cancel*** button to abort it. Once the new filename is supplied and the ***Open*** button is pressed, the previous dialog box is closed, the table shows the default event-record and the name of the new file is shown in the ***Timeline file:*** text-box.

The user may press the ***Load*** button to load a previously saved timeline file, which contains a set of event-records. A dialog box is opened where the user may select a file and press the dialog box's ***Open*** button to continue performing the loading of timeline action, or the ***Cancel*** button to abort it. Once the new filename is supplied and the ***Open*** button is pressed, the previous dialog box is closed, the event-records contained in the selected timeline are shown on the editable table and the name of the loaded file is shown in the ***Timeline file:*** text-box. The dialog box looks for the timeline file in the *SEPSroot / Data / Input / UserInputFiles / Timeline /* folder.

When a set of event-records have been configured or edited the user may save them to file pressing the ***Save*** button (overwrite the current timeline file) or the ***Save as*** button (save the event-records in a new timeline file). In this last case, a dialog box is opened where the user may select a file and press the dialog box's ***Save*** button to continue performing the saving of the event-records action, or the ***Cancel*** button to abort it. Once the new filename is supplied and the ***Save*** button is pressed, the previous dialog

box is closed, the event records are saved in the file in ASCII format and the new filename is shown in the ***Timeline file:*** text-box.

The user may press on the ***Delete*** button in order to remove physically the selected timeline file from the hard disk, and clean the editable table. This action displays a confirmation dialog (Figure V.25) to assure the deleting execution.

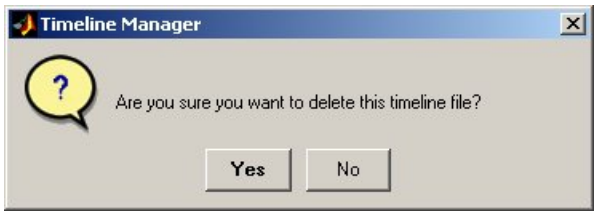


Figure V.25.Confirmation window for deleting a timeline file

The user may press on the ***Accept Timeline*** button in order to update the main SEPS GUI with the selected timeline file in the Timeline Manager windows. This action closes the Timeline Manager windows.

Finally the user may press on the ***Close*** button to close the Timeline Manager windows without updating the selected timeline file.

Note*: The "Save as" action (using a different name) does not imply that the previously selected timeline file for the current scenario is updated. The Timeline Manager should be understood like an auxiliary and independent tool, which is to be used for viewing, changing, updating or selecting timeline files out of the current scenario. Only in the case of the user presses on the "Accept Timeline" button, the selected timeline file in the Timeline Manager windows is also selected as the timeline file for the current scenario and therefore this action updates the main SEPS GUI with the selected timeline file name.*

The timeline can also be edited with any text editor (see below timeline file corresponding to fig. V.24)

```
%%                                                TIMELINE FILE
%%
%% Each row defines an event-record to be simulated. An event-record is a set of events
%% executed periodically from a start time and separated a same time step.
%%
%% 1 column: Flag to indicate if the event record is to be computed (1 Yes, 0 No)
%% 2 column: Start time from T0 (Initial time) for the event-record (in seconds)
%% 3 column: Time step between events of the event-record (in seconds)
%% 4 column: Number of events of the event-record.
%% 5 column: "Snapshot" computation for each event of the event-record (1 Yes, 0 No)
%% 6 column: "Calibration Noise injection" computation for each event of the event-record (1 Yes, 0 No)
%% 7 column: "G-Matrix" computation for each event of the event-record (1 Yes, 0 No)
%% 8 column: "Flat Target Transformation" reference data computation for each event of the event-record (1 Yes, 0 No)
%%
%% NOTES:
%%            1.- The Initial Time (T0) is selected in the main GUI, the timeline is not dependent of the
%%                initial time.
%%          2.- The events which have the G-Matrix computation enable (7 column), only will be executed
%%                if the G-Matrix CGS Image Reconstruction is selected in the main GUI
%%          3.- The events which have the "Flat Target Transformation computation enable (8 column),
%%                only will be executed if the Use fo Flat Target Transformation is selected in the main GUI
%%          4.- If the Flat Target Transformation is enable (8 column) and also it is selected
%%                in the main GUI, the SKY-POINTING will be automatically activate for these events.
%%          5.- If in the first (temporally) event, it is not enabled the calibration events, and in the GUI
%%                the calibration files are not inherited from a previouly executed scenario, then automatically
%%                the SW forces to perform a calibration at the first event (in order to obtain the calibration
%%                files) corresponding to the flag activated in the GUI (G-Matrix, FTT, ...)
%%
%%-----------------------------------------------------------------------------------------------------------------
%%        Use     |  Start time   |  Time Step  | Number Events |  Snapshot | Noise Injection | G-Matrix  | FFT Init     |
%%-----------------------------------------------------------------------------------------------------------------
             1         0.00            4.80            1              0              1              0              0
             1         4.80            2.40            1              1              0              0              0
```

and event lines can be cut and pasted, which may be simpler and faster to create long and complex timelines.

CHAPTER VI. MISSION SIMULATION

Once the scenario to be simulated has been defined, only some more steps are needed before launching a simulation. These are the setting of the type of seed that is to be used in random generations, and also the type of output data that the simulator must generate at user's request.

VI.1. Mission Simulation Interface

The user may click on the ***Mission Simulation*** label in the left lower box of the SEPS main window, in order to define the previously mentioned mission simulation parameters. Then, the interface shown in Figure VI.1 is displayed.

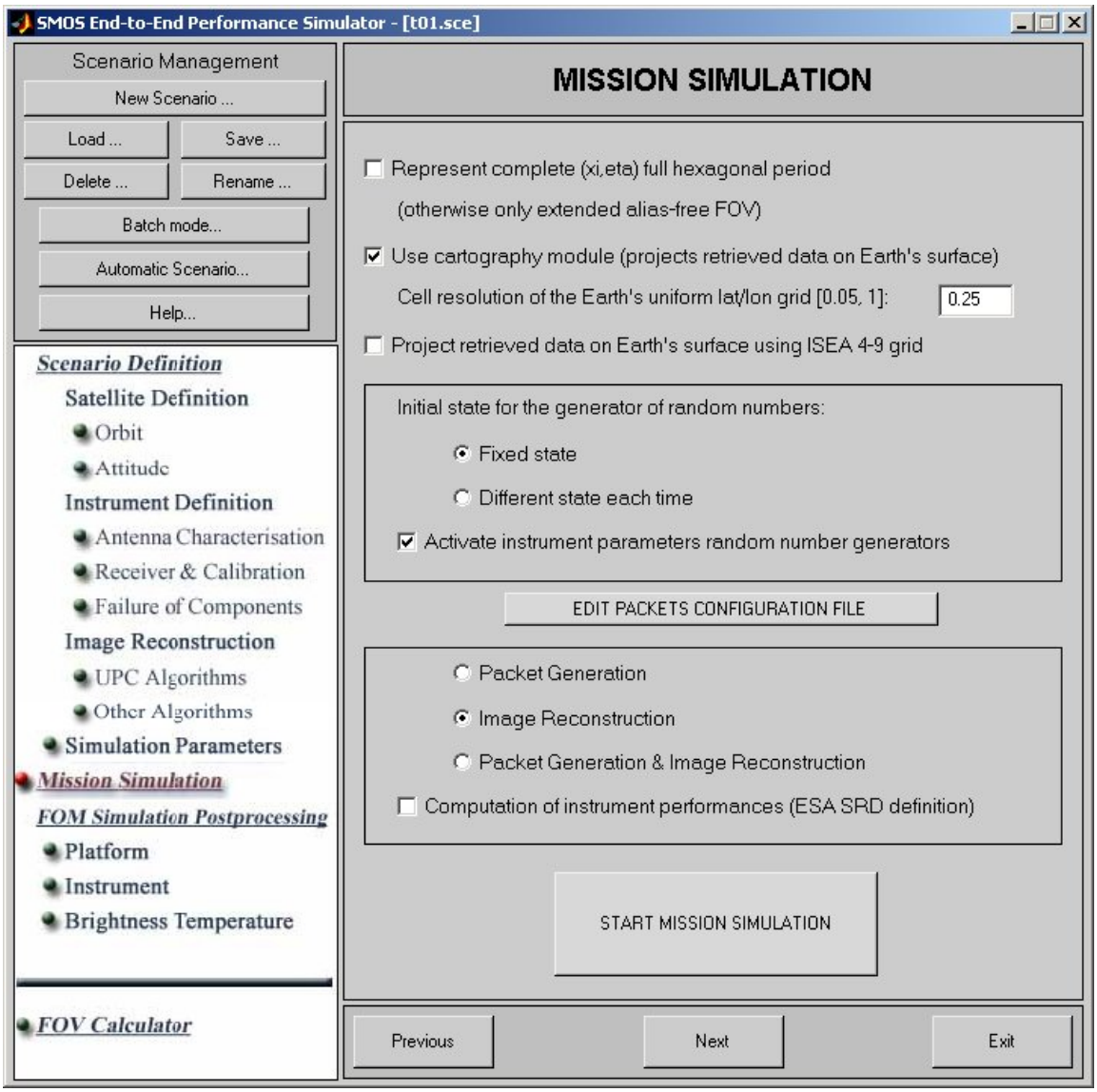


Figure VI.1. Mission Simulation *interface.*

The first parameter is the **Represent complete (xi,eta) full hexagon period.** When this option is selected the whole (ξ,η) hexagonal period is shown, including the alias regions. If a FFT is used, the image shows directly the overlap of aliases. If not, the alias regions cannot be so easily interpreted, since a constant Earth is subtracted in the visibility domain, but then the constant value is added back only in the alias-free FOV during the image reconstruction process.

The second parameter available through this interface allows the user to disable the use of the Cartography Module. If the system does not have the MATLAB Mapping Toolbox 1.3 (for Matlab release 13) or 2.0 (for Matlab release 14) installed, then the ***Use cartography module*** option is always disabled, in order not to try to project the reconstructed brightness temperatures scenes on Earth. In the other hand, if the system has the Mapping Toolbox installed but the user does not need to use the projected data on Earth, disabling this parameter may be interesting to gain time and save disk space.

However, if the user's choice is to project the reconstructed data on the Earth's surface, the ***Cell resolution of the Earth's latitude / longitude grid*** becomes enabled. This parameter allows the user to change the cell resolution of the projected grid on Earth. The default value is 0.25º x 0.25º for each cell, but the range is between 0.05º and 1º. The user may take into account the resolution of the geophysical parameters' database grids when setting the value for this parameter.

Additionally, the user may check the ***Project retrieved data on Earth's surface using ISEA 4-9 grid*** to project the data over the official latitude/longitude SMOS grid. Since SEPS is expected to run in most PCs, a simple, although not the optimal one, nearest neighbour interpolation is used to project over the Earth.

The following parameter to be set is the ***Initial state for the generator of random numbers***: the SEPS v4.0 code includes a certain number of calls to a random number generator, in order to model the system's unpredictable imperfections, but the user may need these random numbers not to change. That is, not to be actually random but repeatable, for example running different simulations with different values for a given input parameter, in order to test its influence in the system's performance. Then, the user may select the ***Fixed state*** radio button to use always the same random generator's seed, and hence effectively make this generator's outputs to be always identical. Otherwise, the user may click the ***Different state each time*** radio button to obtain actual random outputs from the random generator. Moreover, the user may check the ***Activate instrument parameters random number generators*** check-box to generate random numbers in the instrument model.

Through the *Mission Simulation* interface, the user can decide what to do with the visibility data obtained from the simulation computations. Two possible uses can be given to this data, either to generate science and ancillary data packets from the visibility data, or the later use of a reconstruction algorithm to retrieve brightness temperature images. For this purpose, three radio buttons are available to make the choice:

- Selecting the ***Packet Generation*** radio button, the science and ancillary data packets are generated and stored, following the format described in the PLM packet format (see ref. SO-TN-CASA-PLM-0279 and ref. SO-TN-CASA-PLM-0549).
- Selecting the ***Image Reconstruction*** radio button leads to apply the chosen algorithm to retrieve the brightness temperature images;
- Finally, selecting the ***Packet Generation & Image Reconstruction*** radio button, both types of data are retrieved.

The user has also the possibility of editing the Packets Configuration File. Clicking the ***EDIT PACKETS CONFIGURATION FILE*** push button, the MATLAB file editor is automatically launched making the *Packets.txt* file available to be changed by the user, as it can be seen in Figure VI.2.

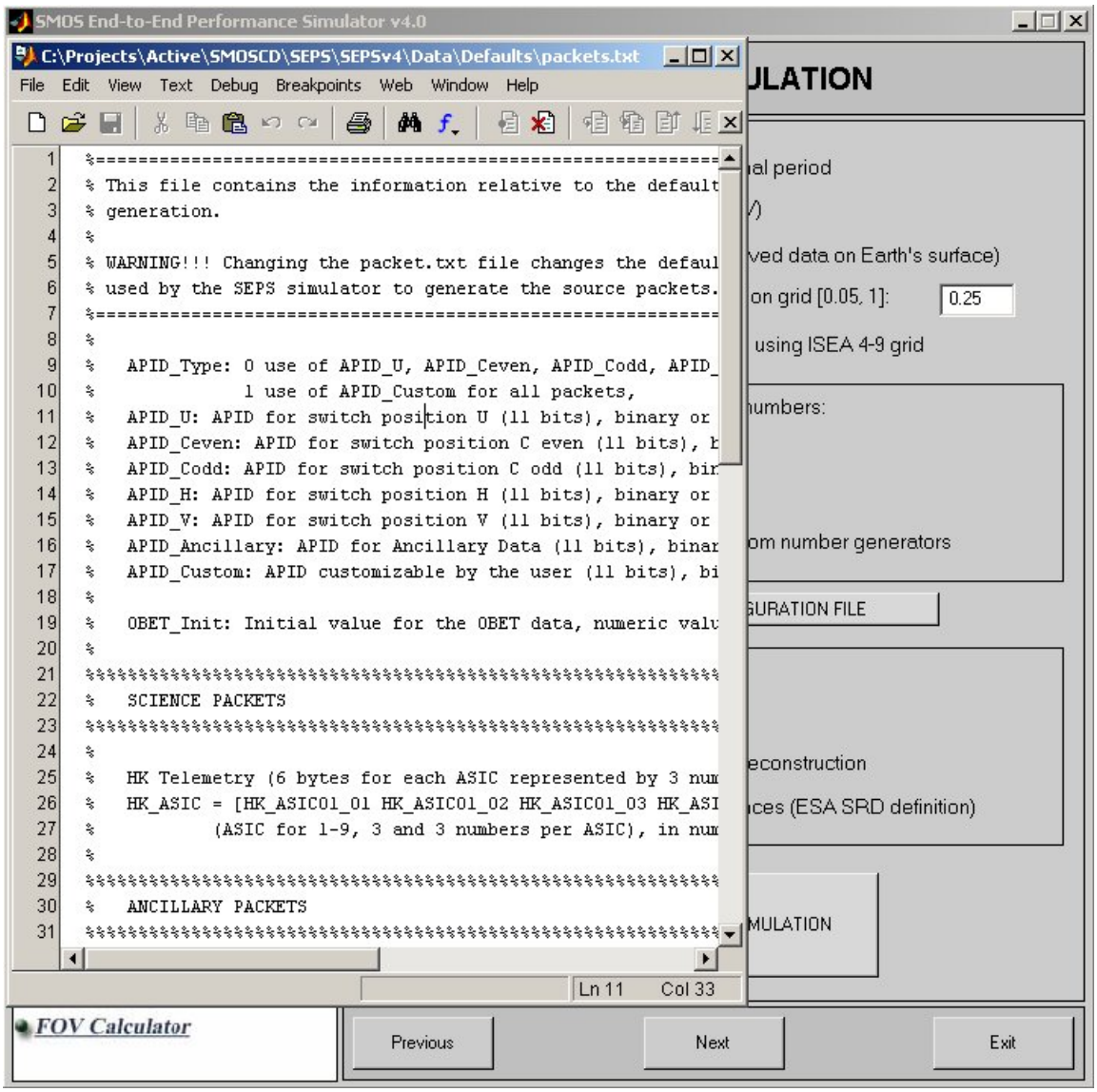


Figure VI.2. Clicking the EDIT PACKETS CONFIGURATION FILE *button launches the MATLAB Editor to allow the user modify the science and ancillary packets format.*

Once the last parameters needed to define a simulation, and having defined the scenario to be computed –as seen in chapter V-, the user may launch a simulation pushing the ***START MISSION SIMULATION*** button.

***IMPORTANT**: once a scenario simulation is launched, the user may not try to modify any section of the SEPS environment. MATLAB will try to perform these modifications during the simulation execution and may abort the simulation.*

VI.2. Monitoring a simulation

Once the user has launched a simulation, interaction is not needed until the end of the computations. However, the user may consider interesting to know what is happening during the current scenario simulation. To achieve this, SEPS uses MATLAB capabilities to communicate the user the simulation progress. This section gives an overview of the simulation progress of a typical simulation. Of course, each simulation is different from the others and this section does not pretend to illustrate each message displayed through the MATLAB environment.

The simulation progress is basically illustrated through text messages displayed in the MATLAB Command Window, as it is seen in Figure VI.3.

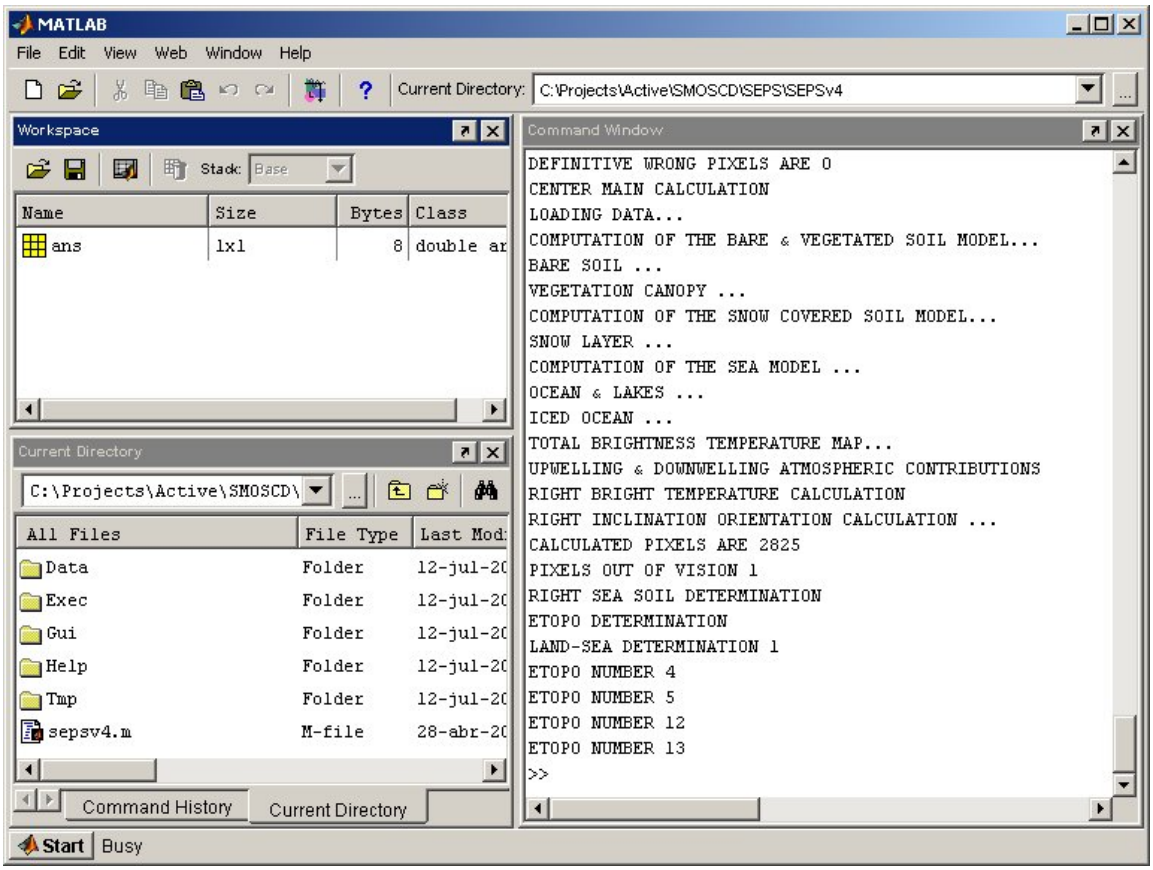


Figure VI.3. MATLAB Command Window appearance after launching a simulation. Messages displayed through it inform the user about the simulation progress.

However, other graphic tools are also used to illustrate the simulation progress. After completing the execution of the brightness temperature generator module, the images arriving to the antenna frame are shown in a MATLAB figure as it is seen in Figure VI.4. Both X and Y polarizations images are drawn, with a scale color bar illustrating the range of brightness temperatures of the images plotted. Notice the figure's name –upper part of the figure- informs about the number of snapshot in the simulation currently being computed.

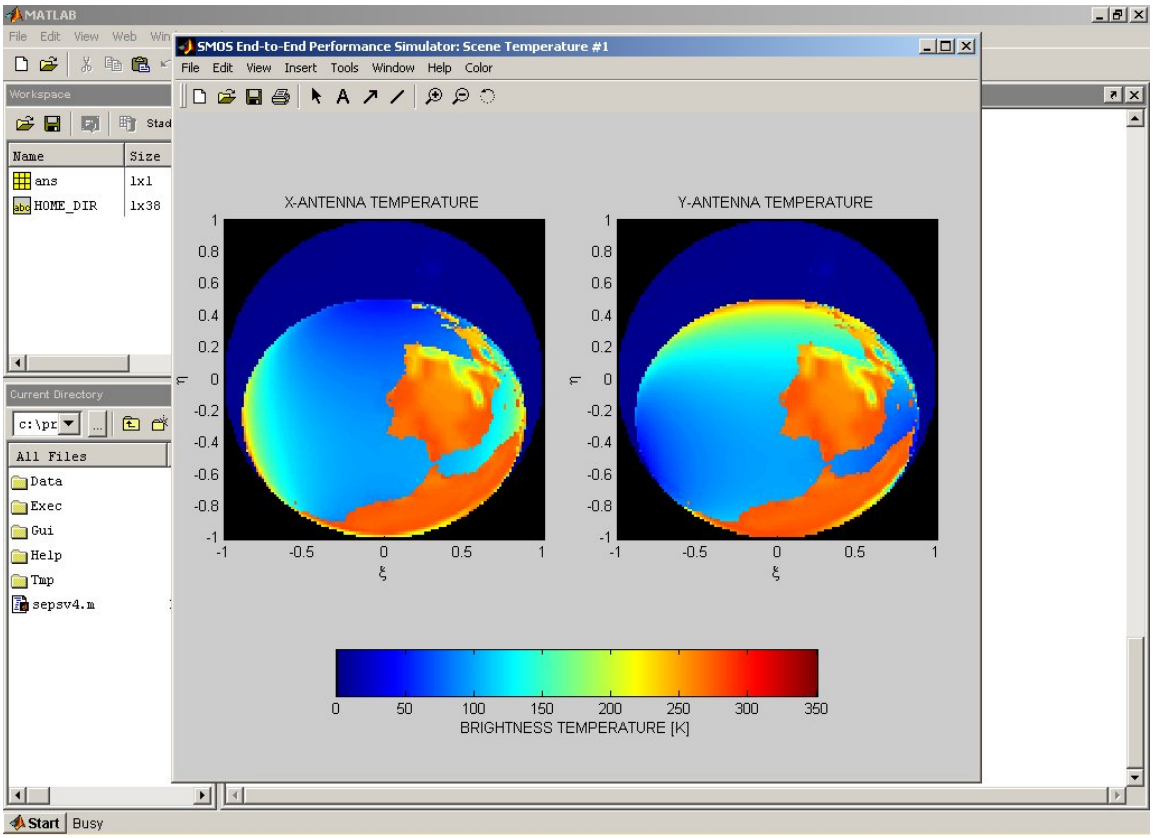


Figure VI.4. Figure showing the brightness temperature images at the antenna frame, for both Vertical and Horizontal polarizations. The figure's name contains the number of snapshot currently being simulated.

After the brightness temperature generator is executed for the current snapshot, the Observables Generator for obtaining visibility samples is executed. The progress of these computations is shown using waitbars, which inform of the already computed work done in that concrete step of the simulation. Figure VI.5 shows an example for the computation of the visibility samples. Notice that the percentage of computed work is shown in the bar's upper part, and the type of action is described inside the figure.

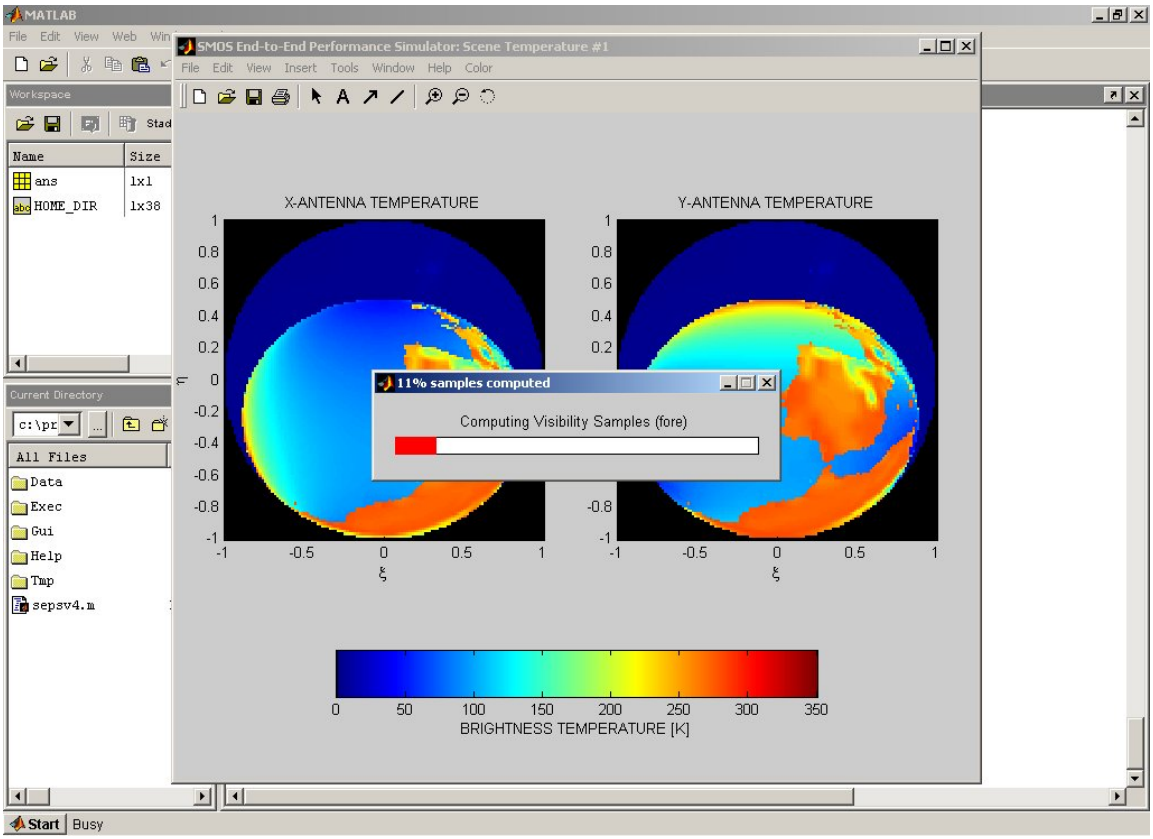


Figure VI.5. Use of waitbars to show a module computation progress. The figure's name –upper part of the waitbar figure- shows the percentage of work done, while the text label inside the figure informs about the action currently being executed

Waitbars are used in various modules of the simulator, among others and in chronological order:

- Antenna temperatures computation (prior to the observables generator);
- Visibility samples;
- Auxiliary visibility samples;
- Antenna patterns computation (image reconstruction);
- Extended-clean image reconstruction iteration, if this algorithm is used. If others, computation progress is displayed through the MATLAB Command Window;
- Projection of retrieved data on Earth's surface, if selected (see section VI.1).

This process repeats for each snapshot of the simulation. At the end of the simulation, the retrieved data is saved to disk, and a figure like Figure VI.6 is displayed, showing the statistics of the retrieved images in the simulation, particularly the biases and accuracies between the reconstructed images -both Y / X polarizations- and the original antenna -obtained from the *Brightness Temperature Generator* module)- and the ideal FFT brightness temperature images. This same figure is also displayed when calling the *Cosines Domain* tool (see *FOM Simulation Postprocessing* chapter to obtain more information on this statistics figure). It must be said that obviously this figure is only presented when the image reconstruction modes are activated (see section VI.1).

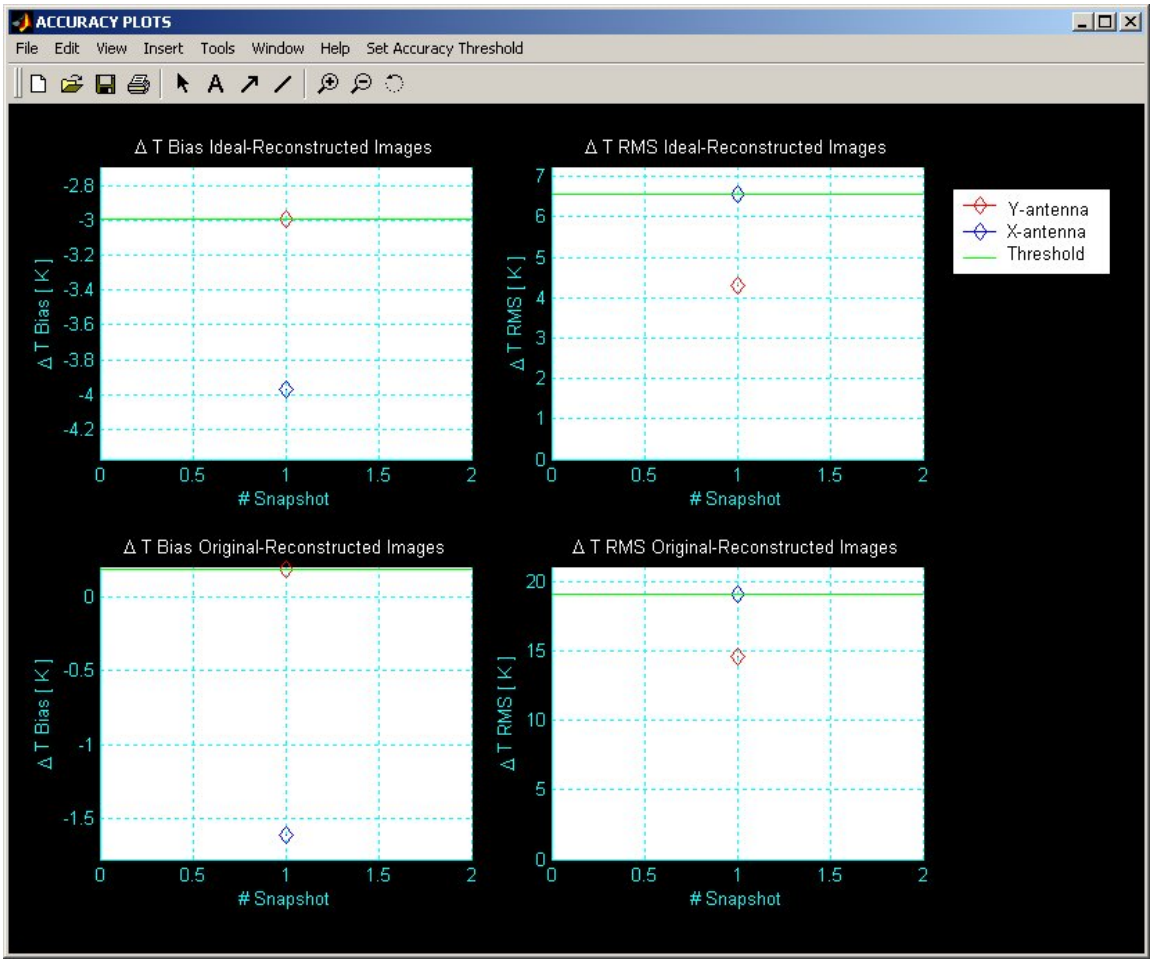


Figure VI.6. Statistics figure shown at the end of the simulation, containing the biases and accuracies evolution for all the snapshots of the simulation.

Once the simulation has been executed, the message shown in Figure VI.7 is presented so as to warn the user about it.

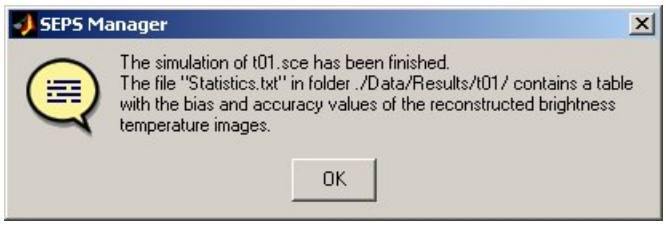


Figure VI.7. Message presented at the end of the simulation so as to warn the user about the finalization of it.

CHAPTER VII. FOM SIMULATION POST-PROCESSING

After executing a scenario simulation (see Chapters V and VI for details on the *Scenario Definition* and *Mission Simulation* section, necessary to configure and run one scenario simulation), some tools are available to the user for evaluating the results.

This chapter reviews in detail the use of the *FOM (Figures Of Merit) Simulation Postprocessing* tools, which are available for examining the output-simulated data. Three groups of tools are available:

- The ***Platform FOM***, concerning the evolution of most of the important orbital parameters. The tools provided in this group are explained in section VII.1.
- The ***Instrument FOM***, for examining the radiometer's performance during the simulation, concerning aspects such as the system temperatures, and the drifts in receiver's frequency responses, among others. The use of the tools available in this group is detailed in section VII.2.
- The ***Brightness Temperature FOM***, which is a group of tools for projecting the measured brightness temperature images, as well as some geophysical parameters needed for their computation, on certain cartographic maps and in the cosines domain. Section VII.3 reviews theses tools.

The user may note that when any Figure Of Merit is displayed, automatically the JPEG image representing the figure is generated and stored in the *SEPSroot/Data/Results/ScenarioName/* folder -being *SEPSroot* the folder where the SEPS v4.0 is installed, and *ScenarioName* the name given by the user to the simulated scenario. Other MATLAB default tools for figures are available, such as the *Save* icon , the *Print* icon , etc… Please refer to the MATLAB documentation to know more about these utilities.

VII.1. Platform Figures Of Merit

This set of tools allows the user to evaluate the evolution of the orbital characteristics for an already computed simulation. The user must press the ***Platform*** label in the *FOM Simulation Postprocessing* to display the interface shown in Figure VII.1.

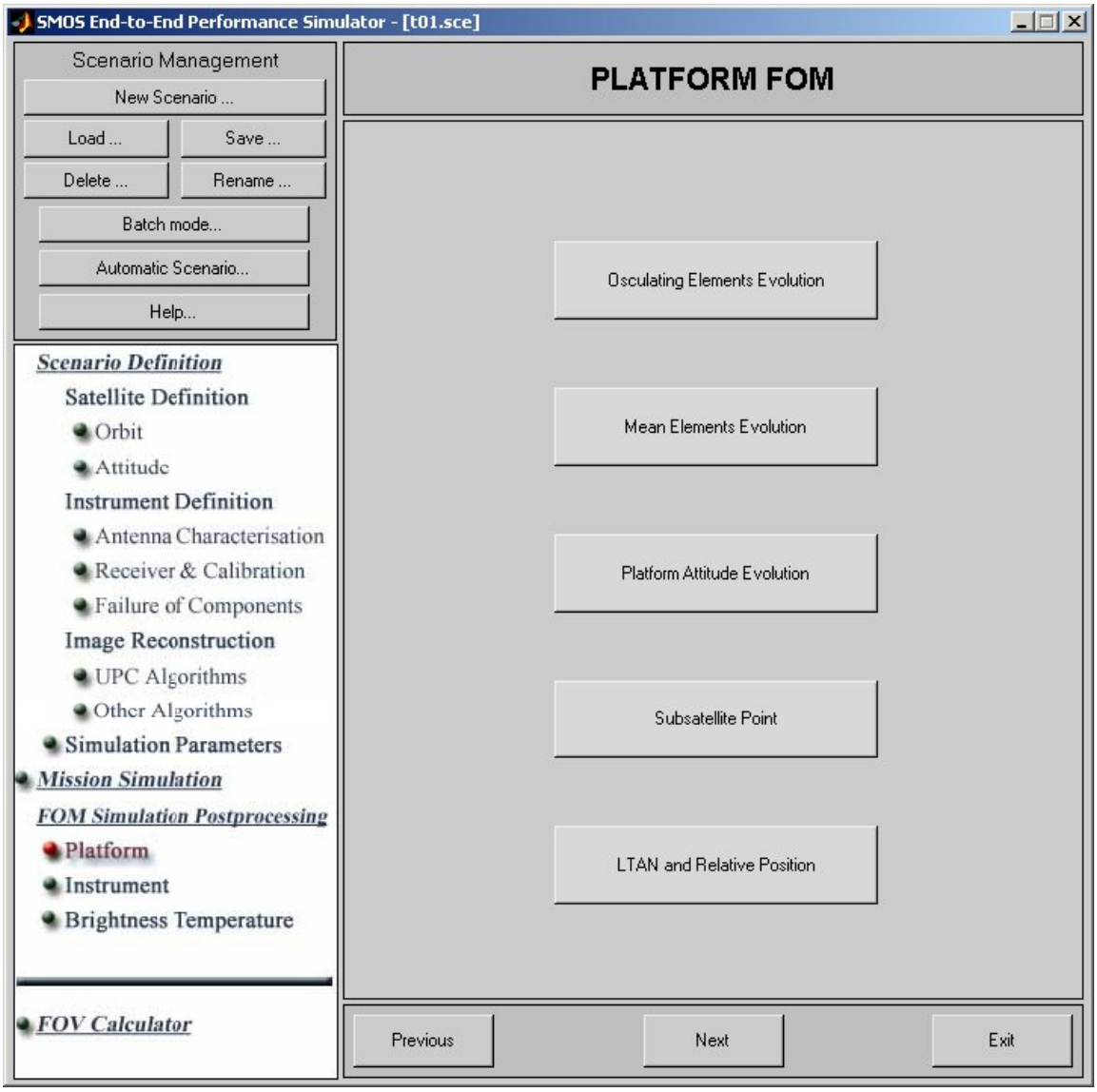


Figure VII.1. Platform FOM *interface, to plot the evolution of the orbital parameters of an already simulated scenario.*

As it can be observed, five tools are available:

- two for monitoring the evolution of the platform's position and velocity (***Osculating Elements Evolution*** and ***Mean Elements Evolution***), explained in section VII.1.1 and VII.1.2;
- one for the attitude of the satellite (***Platform Attitude Evolution***), detailed in section VII.1.3;
- one for the platform's position in the map (***Subsatellite Point***), reviewed in section VII.1.6; and finally
- one for plotting the evolution of the local time and the platform's relative position (***LTAN and Relative Position***), explained in section VII.1.7.

For further information on the platform figures of merit, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.55).

VII.1.1. Osculating Elements Evolution

The classical osculating elements for an already performed simulation can be plotted by pressing the **Osculating Elements Evolution** push button in the *Platform* interface. Figure VII.2 shows the resulting plot.

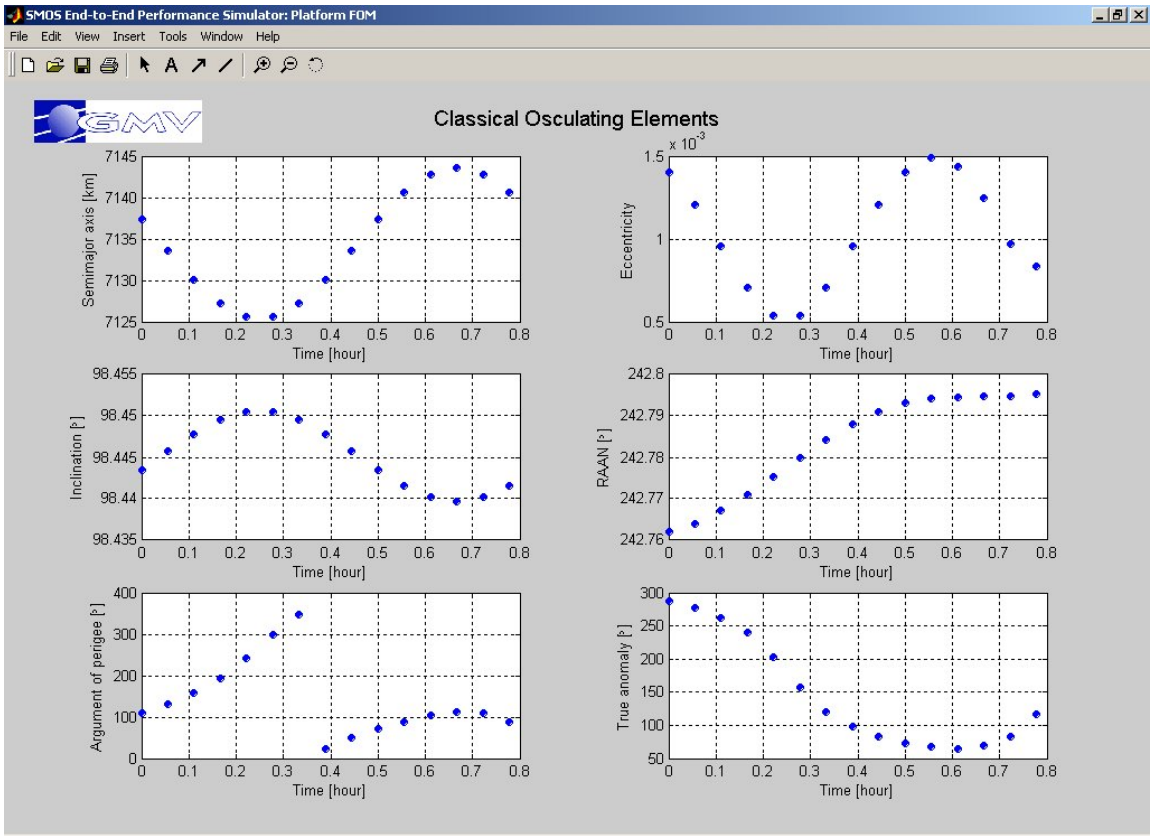


Figure VII.2. Classical Osculating Elements *plot for displaying the evolution of this type of orbital parameters along the computed simulation.*

The displayed characteristics (each blue circle shows the osculating elements corresponding to an event), from left to right and from up to down are:

- evolution of the semi-major axis of the ellipse described by the platform in its orbit around the Earth. Units are km;
- evolution of the platform orbit's eccentricity;
- the evolution of the orbital inclination, in sexagesimal degrees;
- the orbit's right ascension of the ascending node, in sexagesimal degrees;
- the argument of the perigee, in sexagesimal degrees;
- and the true anomaly, in sexagesimal degrees.

For further information on the computations of these parameters the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.7 *MS1: Mission Simulation. Orbital Propagator Module*).

VII.1.2. Mean Elements Evolution

To display the evolution of the mean orbital elements, the user may press the ***Mean Elements Evolution*** push button in the *Platform* interface. Then a plot as the one seen in Figure VII.3 is presented.

The figure displays six plots with the classical mean elements evolutions with time (each blue circle shows the mean elements corresponding to an event). From left to right and from up to down, the plots display:

- the evolution of the semi-major axis of the ellipse described by the platform in its orbit around the Earth, in km;
- the evolution of the orbit's eccentricity;
- the evolution of orbit's inclination, in sexagesimal degrees;
- the evolution of right ascension of the ascending node, in sexagesimal degrees;
- the evolution of the argument of perigee, in sexagesimal degrees;
- and the evolution of the eccentricity vector.

For further information on the computations performed to achieve these output parameters, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.7 *MS1: Mission Simulation. Orbital Propagator Module*).

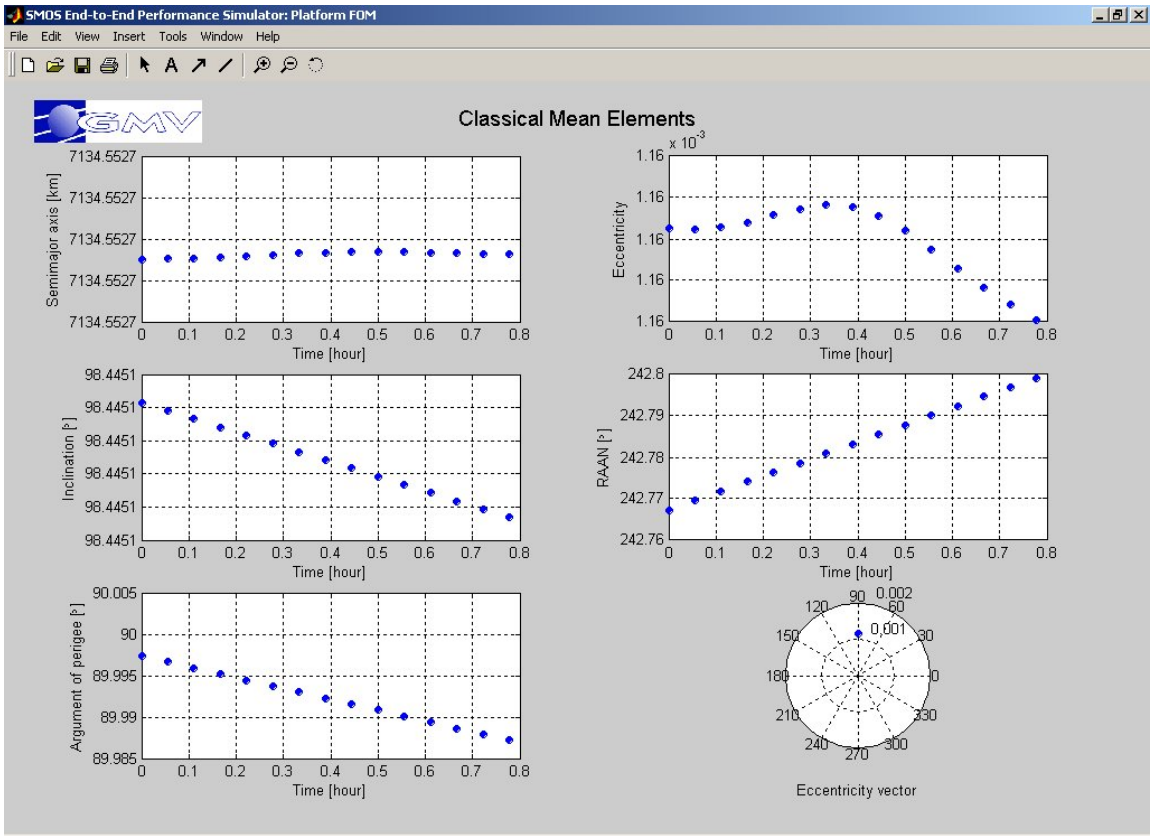


Figure VII.3. Classical Mean Elements Evolution *plot for displaying the evolution of this type of orbital parameters along the computed simulation*

VII.1.3. Platform Attitude Evolution

The platform's attitude parameters can be plotted by pressing the **Platform Attitude Evolution** button in the *Platform* interface. A plot like the one seen in Figure VII.4 is opened.

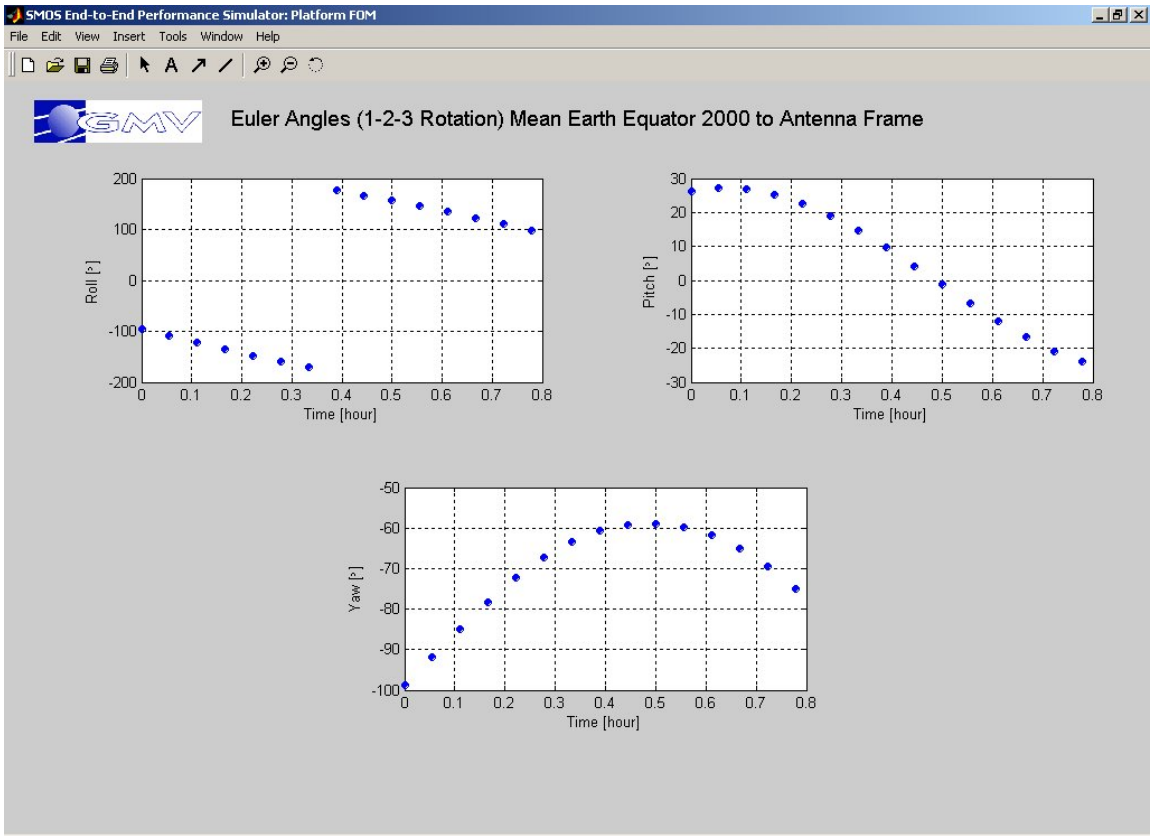


Figure VII.4. Platform Attitude Evolution *plot for displaying the evolution of the Euler angles to transform the inertial reference frame to the body reference frame.*

The evolution of the three Euler attitude roll, pitch and yaw angles (in 1-2-3 sequence) evolutions with time are plotted (each blue circle shows the attitude angles corresponding to an event). These angles characterize the transformation matrix between the Mean Earth Equator MJD2000 reference frame and the Antenna reference frame. For further information on the computations of the platform's attitude the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.9 *MS1.3: Mission Simulation. Orbital Propagator Module. Attitude Profile Generator*).

VII.1.4. Sub-satellite Point

Pushing the **Sub-satellite Point** button in the *Platform* interface, the user opens a figure displaying a world map where the movement of the platform's nadir point is shown

by red circles representing the subsatellite points for each event, as it is seen in Figure VII.5.

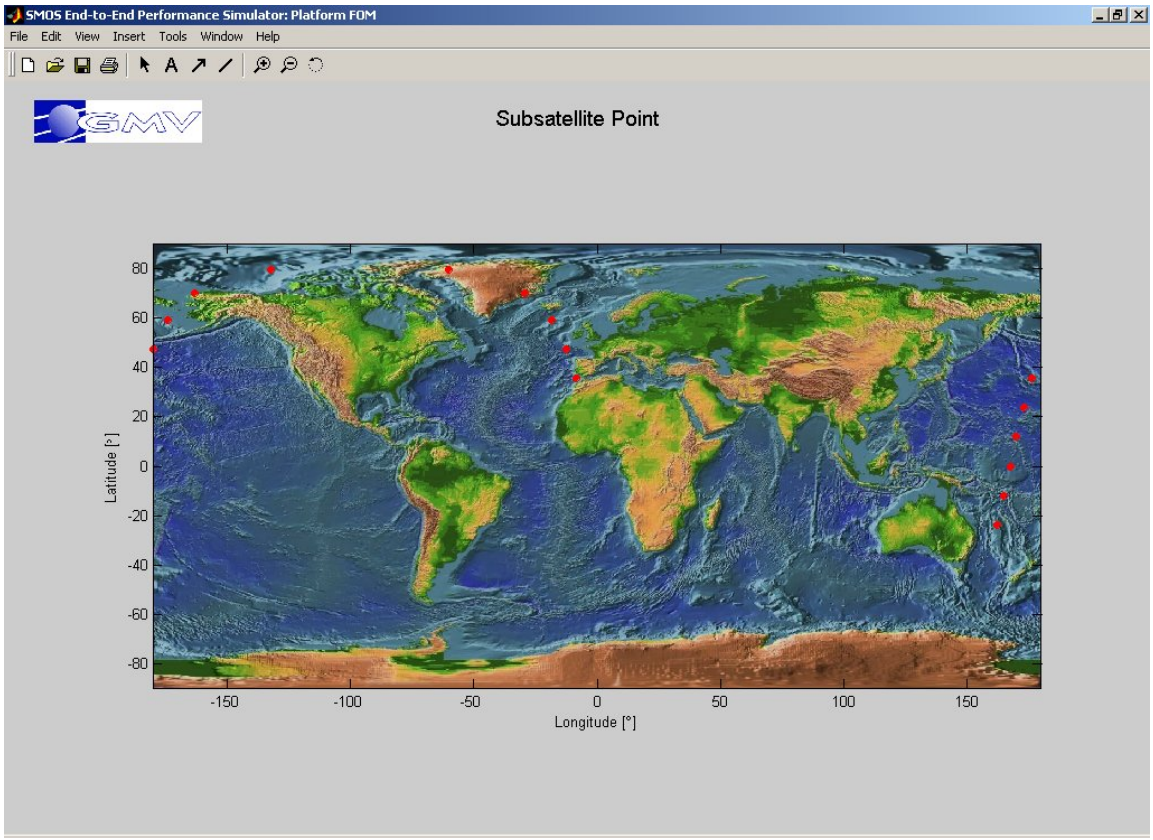


Figure VII.5. Sub-satellite Point Evolution *plot.*

For more information on this topic the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.10 *MS1.4: Mission Simulation. Orbital Propagator Module. Auxiliary Geometric Calculator*).

VII.1.5. LTAN and Relative Position

Pushing the ***LTAN and Relative Position*** button in the *Platform* interface, the user opens a figure displaying three plots, as it is seen in Figure VII.6.

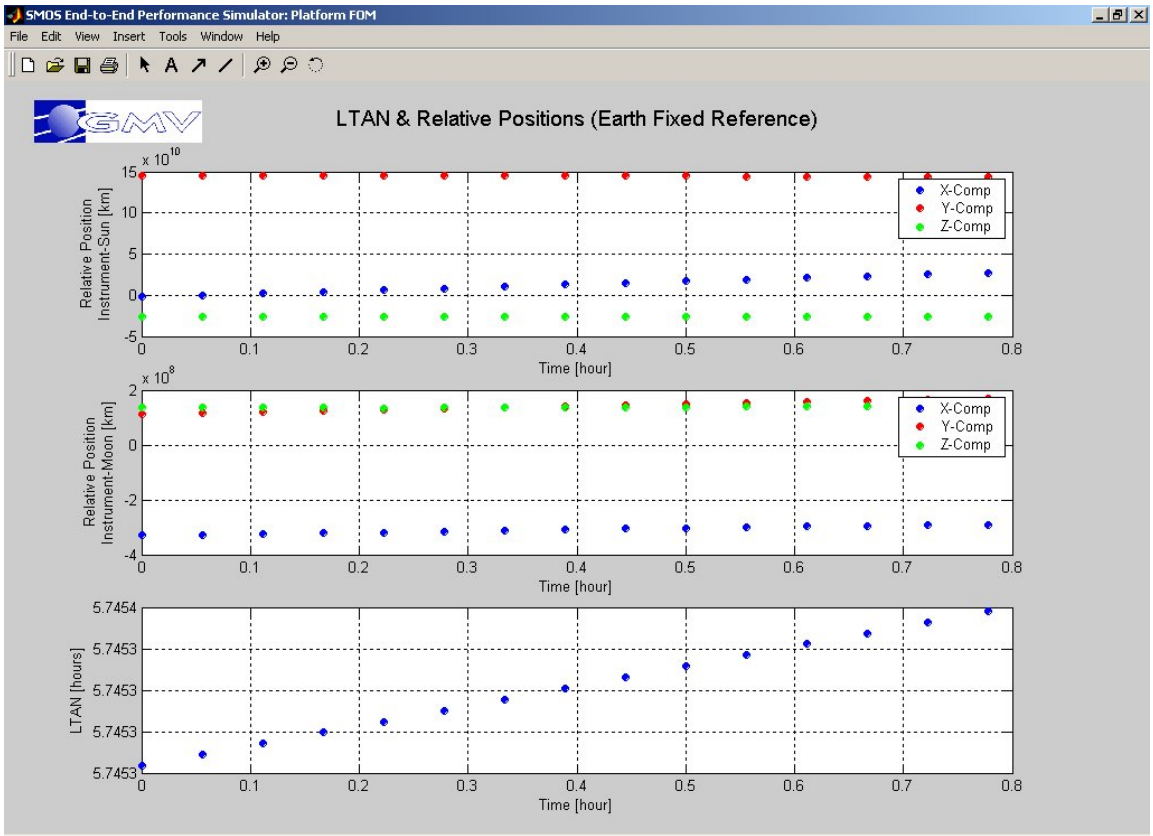


Figure VII.6. LTAN and Relative Positions Evolution *plot*

The figure above shows the evolution of the relative position –decomposed in X, Y, Z components- between the Instrument and the Sun, and also between the Instrument and the Moon. Also, the third plot shows the local time of the ascending node (LTAN) for each position of the platform along the simulation. Each circle shows the LTAN and relative SC-Sun/Moon positions corresponding to an event.

VII.2. Instrument

Through the *Instrument* FOM tool, the user has the capability of evaluating the evolution of the SMOS instrument main characteristics along the simulation. To do so, the user may click on the ***Instrument*** label in the *FOM Simulation Post-processing*. This tool consists of five options, which are detailed in the next subsections.

For more information on this topics the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.56 *FOM2: FOM and Statistical Analysis. Instrument FOM*).

VII.2.1. Antenna Patterns

This option allows the user to plot the co-polar and cross-polar module and phase diagrams of a user-selected antenna of the array, for both X and Y polarizations. When the ***Antenna Patterns*** push button is pressed in the *Instrument* menu, the interface shown in Figure VII.7 is presented.

This interface shows the array as it is currently configured through the *Antenna Characterization* and *Failure of Components* interfaces of the *Instrument Definition* section, in the *Scenario Definition* box. That means, taking into account the steering angle orientation, the number of antennas per arm and the number of redundant antennas per arm, and also the LICEF that are failing. The yellow circles represent the antennas, while the circles in cyan color represent the ones which are failing, that cannot be selected. Also the numbers of the antennas in the arms' extremes –first redundant and last normal antenna- are shown, in order to help the user to have a visual idea of the array's orientation.

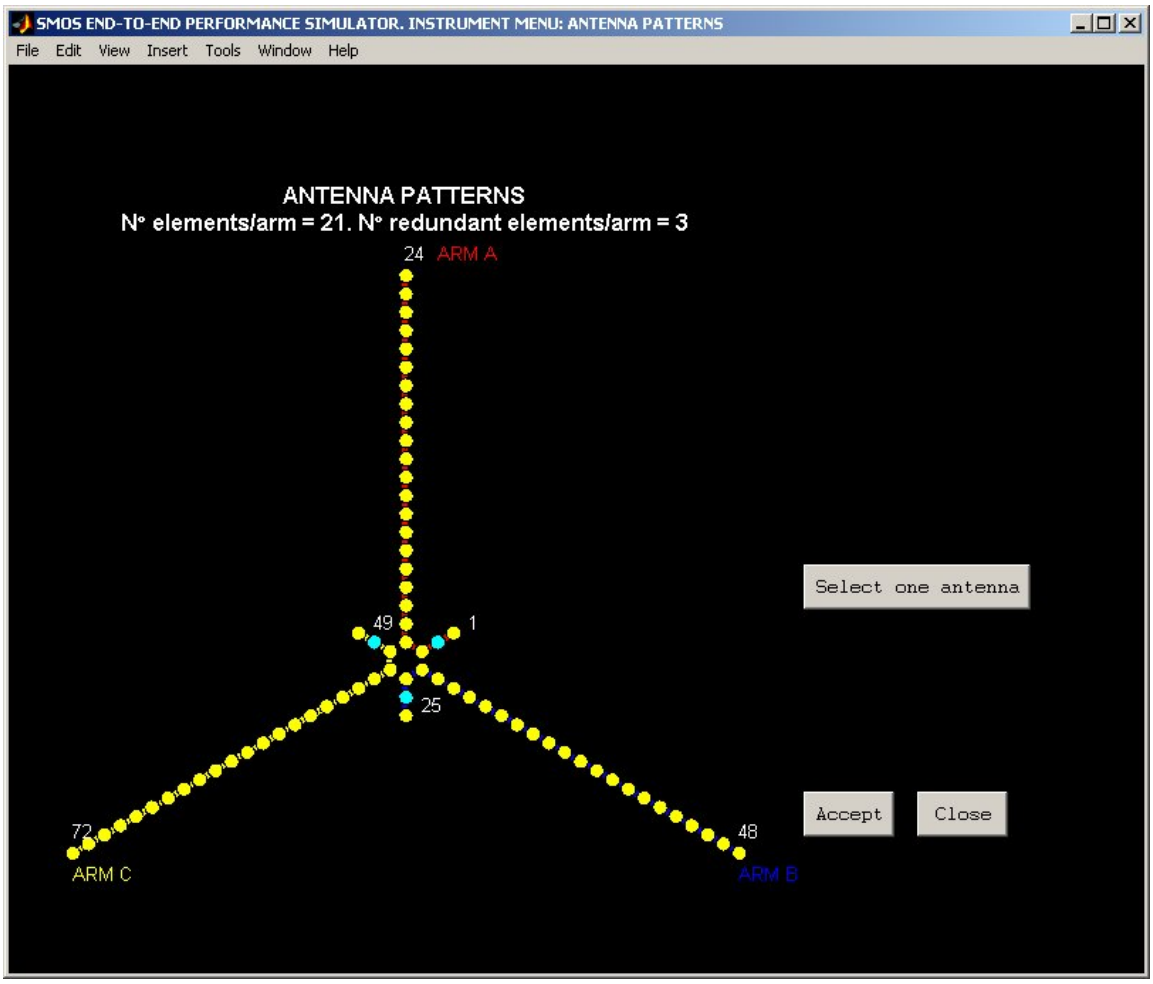


Figure VII.7. Antenna Patterns *interface, called through the* Instrument FOM*. The user must select the antenna whose patterns are to be plotted using this interface*

The ***Select One Antenna*** button must be pushed to choose the device whose antenna patterns are required to plot. Once the referred button is pushed, a cursor like the one displayed in Figure VII.8 becomes available to let the user select the suitable antenna by clicking the left mouse button over the yellow circle that represents the antenna. The user may notice that circles in cyan cannot be selected, as they have been forced to fail through the *Failure of Components* interface in the *Instrument Definition* section.

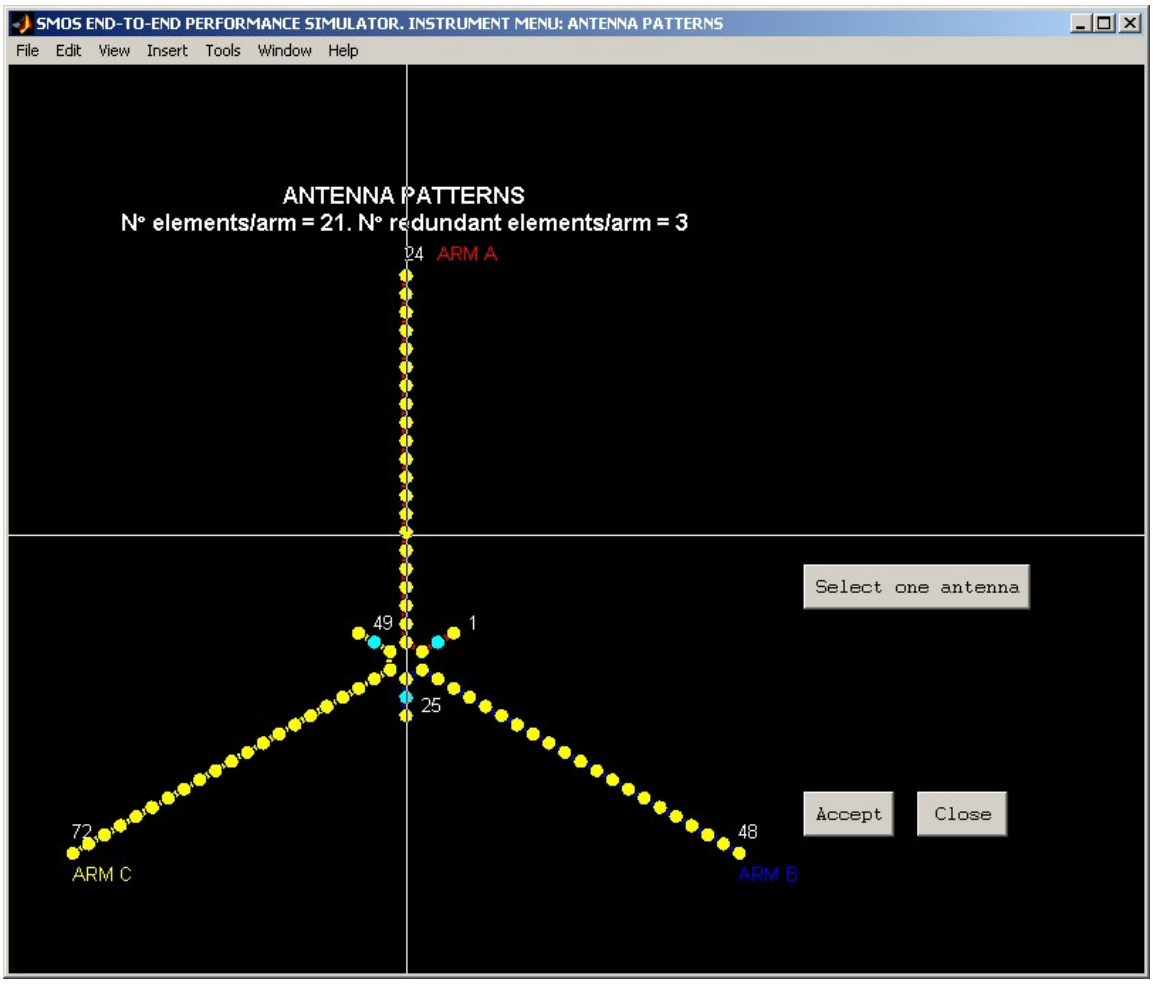


Figure VII.8. Pushing Select one antenna *button, a cursor becomes available to let the user select the antenna whose patterns are to be plotted. Circles in cyan color cannot be selected.*

Once the antenna number is obtained, the user may press the ***Accept*** button to plot the selected antenna patterns. If so, two windows like Figure VII.9 and Figure VII.10 are shown. The patterns correspond to the type of antenna patterns –either based in a mathematical function or loaded from a database- selected in the *Pattern and Thermal* interface in the *Instrument Definition* section. If the mathematical model had been selected, the directivity, ripples, etc… parameters are used.

Figure VII.9 shows four subplots for the co-polar patterns of the selected antenna. The first subplot in the left upper axis is the antenna Y polarization co-polar pattern module. The user may notice that in this forth SEPS release, only the points within the unit

circle –cosines domain- are displayed, for better evaluation of the characteristics. The second subplot in the upper right axis is the antenna Y co-polar pattern phase. The two lower subplots show the same patterns for the X polarization.

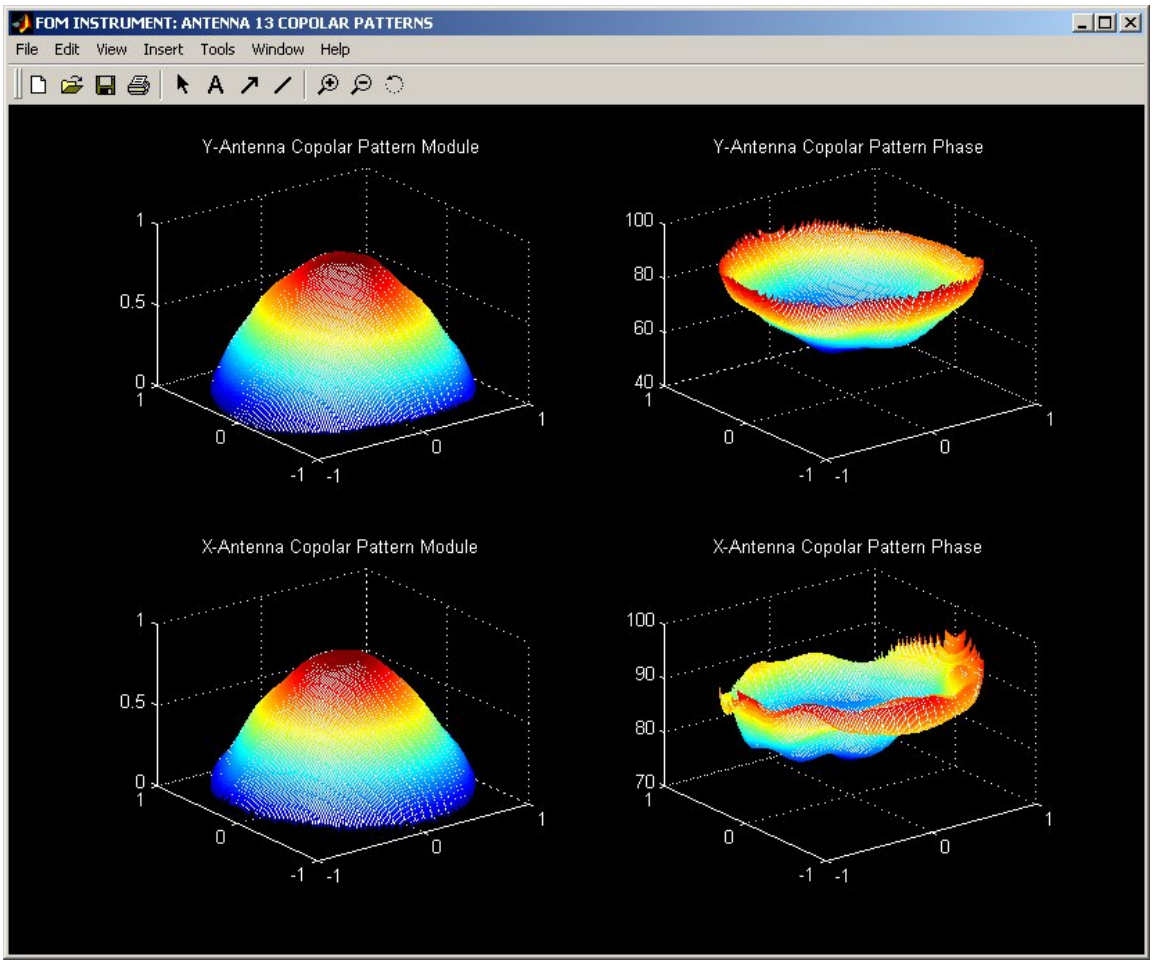


Figure VII.9. The first results figure opened shows the module and phase of the selected antenna co-polar patterns, for both Y and X polarizations.

Figure VII.10 shows four subplots for the cross-polar patterns of the selected antenna. The figure is equivalent to the co-polar one presented in Figure VII.9, but showing the cross-polar patterns for the antenna selected by the user, for both Y and X polarizations.

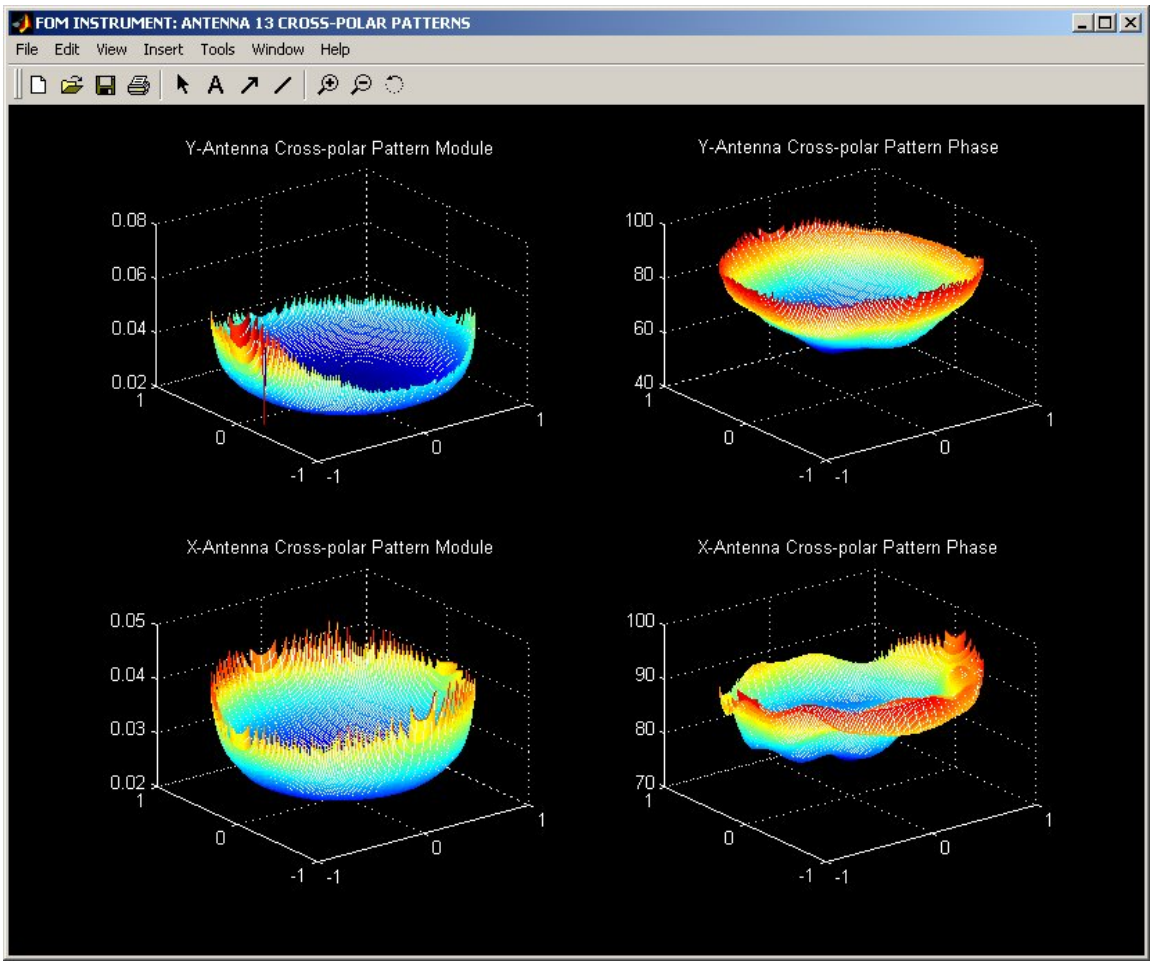


Figure VII.10. The second results figure opened shows the module and phase of the selected antenna cross-polar patterns, for both Y and X polarizations

Once the antenna patterns' figures have been examined, the user can close them pushing the ***Close*** button seen in Figure VII.8.

VII.2.2. Antenna Patterns Differences

This option allows the user to plot the module and phase of the difference between the co-polar and cross-polar diagrams of a user-selected antenna of the array, for both Y and X polarizations. When the ***Antenna Patterns Differences*** push button is pressed in the *Instrument* menu, the interface shown in Figure VII.11 is presented.

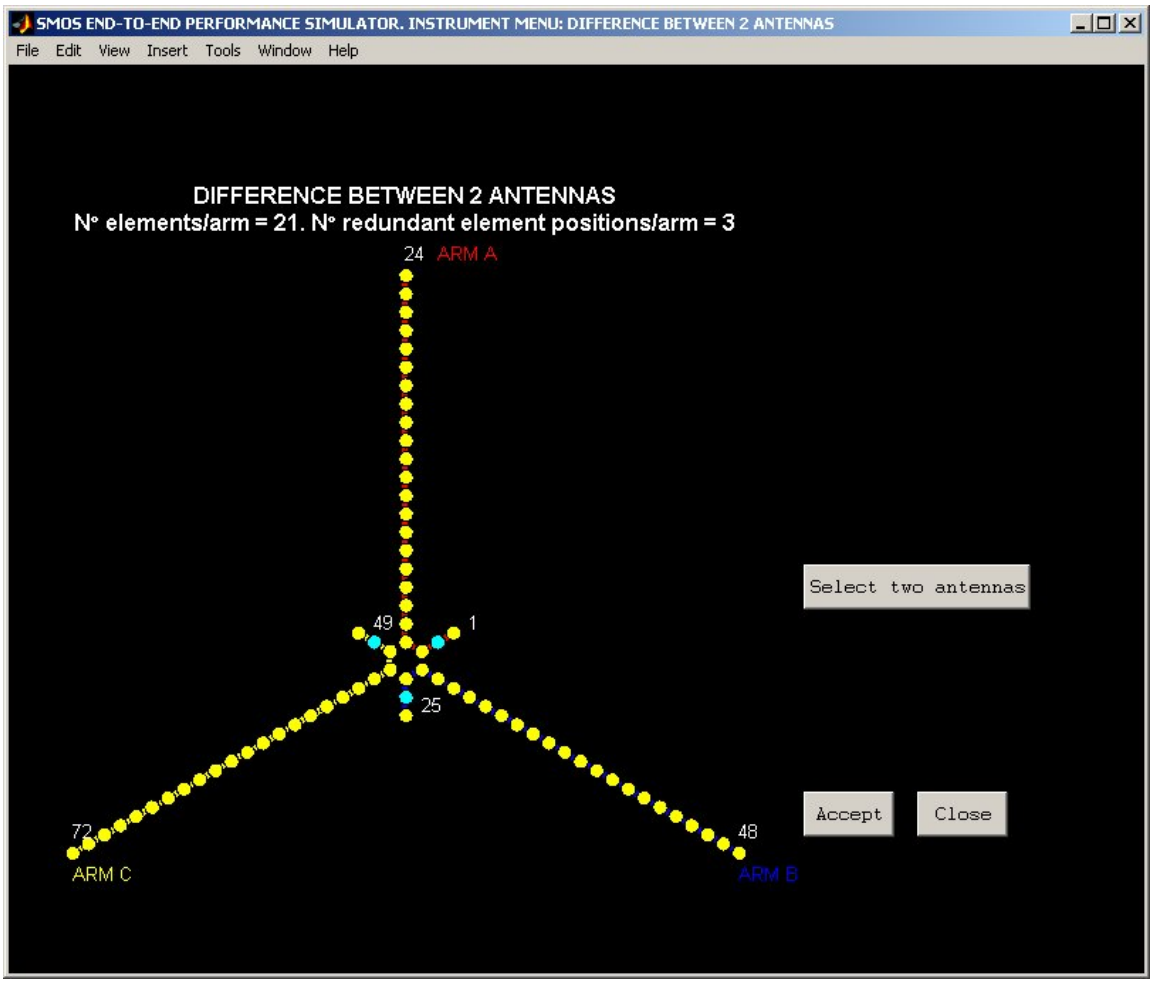


Figure VII.11. Antenna Patterns Differences *interface, called through the* Instrument FOM*. The user must select two antennas whose patterns differences are to be plotted.*

This interface shows the array as it is currently configured through the *Antenna Characterization* and *Failure of Components* interfaces of the *Instrument Definition* section, in the *Scenario Definition* box. That means, taking into account the steering angle orientation, the number of antennas per arm and the number of redundant antennas per arm, and also the LICEF that are failing. The yellow circles represent the antennas, while the circles in cyan color represent the ones which are failing, that cannot be selected. Also the numbers of the antennas in the arms' extremes –first redundant and last normal antenna- are shown, in order to help the user to have a visual idea of the array's orientation.

The ***Select Two Antennas*** button must be pushed to choose the devices whose antenna patterns differences are wanted. Once the referred button is pushed, a cursor like the one displayed in Figure VII.12 becomes available to let the user select the suitable antennas by clicking the left mouse button over two of the yellow circles representing the antennas. The user may notice that circles in cyan cannot be selected, as they have been forced to fail through the *Failure of Components* interface in the *Instrument Definition* section.

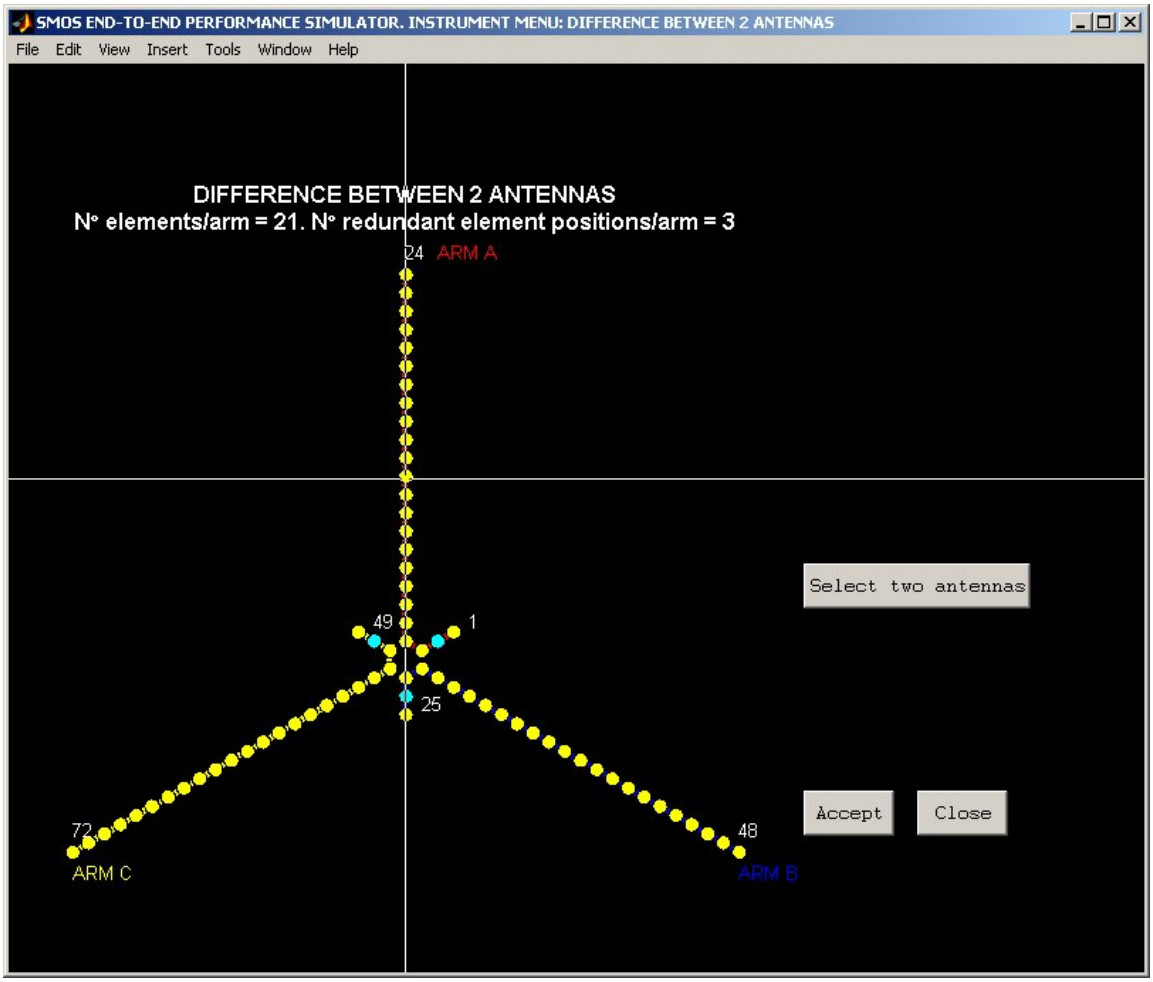


Figure VII.12. Pushing Select two antennas *button, a cursor becomes available to let the user select the two antennas whose patterns differences are to be plotted. Antennas represented by circles in cyan color cannot be selected*

Once the antennas' identifying numbers are obtained, the user must press the ***Accept*** push button to plot the differences between the patterns of the selected antennas.

If so, two windows like Figure VII.13 and Figure VII.14 are shown. The patterns are computed from the input parameters defined in the *Pattern and Thermal* interface in the *Instrument Definition* section. If the mathematical model had been selected, the directivity, ripples, etc… parameters are used.

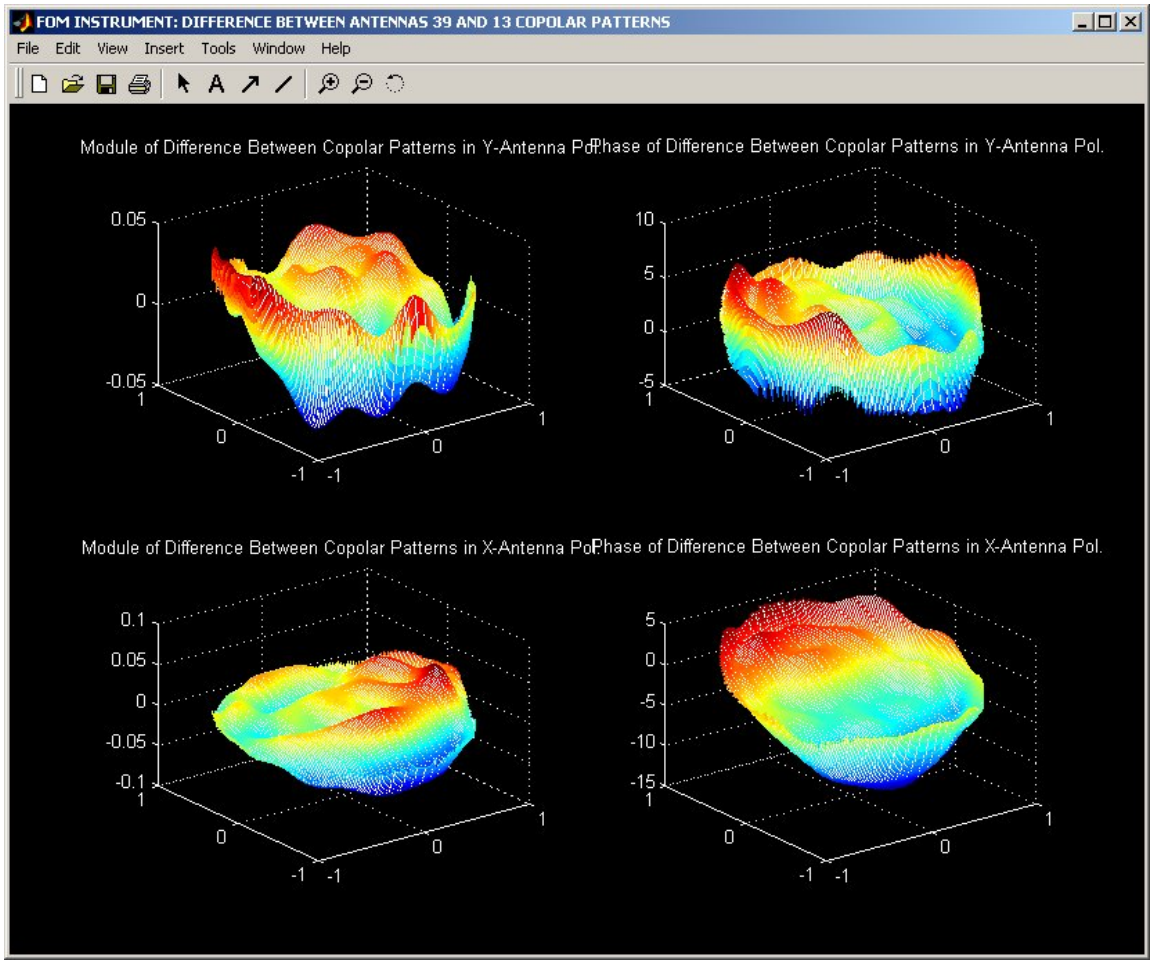


Figure VII.13. The first results figure opened shows the module and phase of the difference between the co-polar patterns of the selected antennas, for both Y and X polarizations.

Figure VII.13 shows four subplots for the differences between the co-polar patterns of the selected antennas. The first subplot in the left upper axis is the module of the difference between the co-polar patterns of the antennas, for Y polarization. The user may notice that in this fourth SEPS release, only the points within the unit circle –cosines domain- are displayed, for better evaluation of the characteristics. The second subplot in

the upper right axis is the phase of the difference between the co-polar patterns of the antennas for the same polarization. The two lower subplots show the same graphics for the X polarization.

Figure VII.14 shows four subplots for the differences between the cross-polar patterns of the selected antennas. The figure is equivalent to the co-polar one presented in Figure VII.13, but showing the differences between the cross-polar patterns, for both Y and X polarizations.

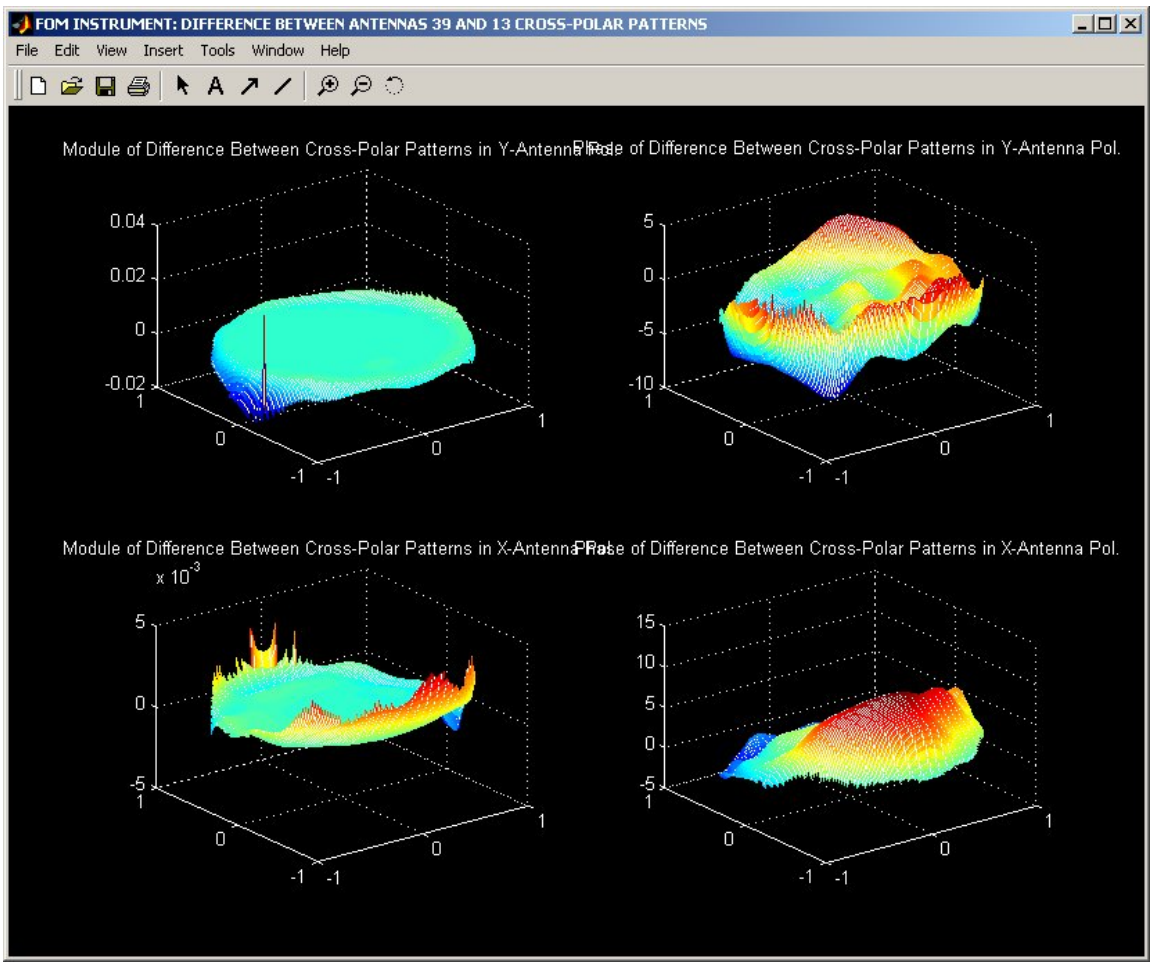


Figure VII.14. The second results figure opened shows the module and phase of the selected antenna cross-polar patterns, for both Y and X polarizations

Once the antenna patterns' figures have been examined, the user can close them pushing the ***Close*** button seen in Figure VII.12.

VII.2.3. Receiver Response

This option allows the user to plot the evolution of the main characteristics of a selected receiver of the array, such as the modules and phases of the channels I and Q of the receiver. When the ***Receiver Response*** push button is pressed in the *Instrument* menu, the interface shown in Figure VII.15 is presented.

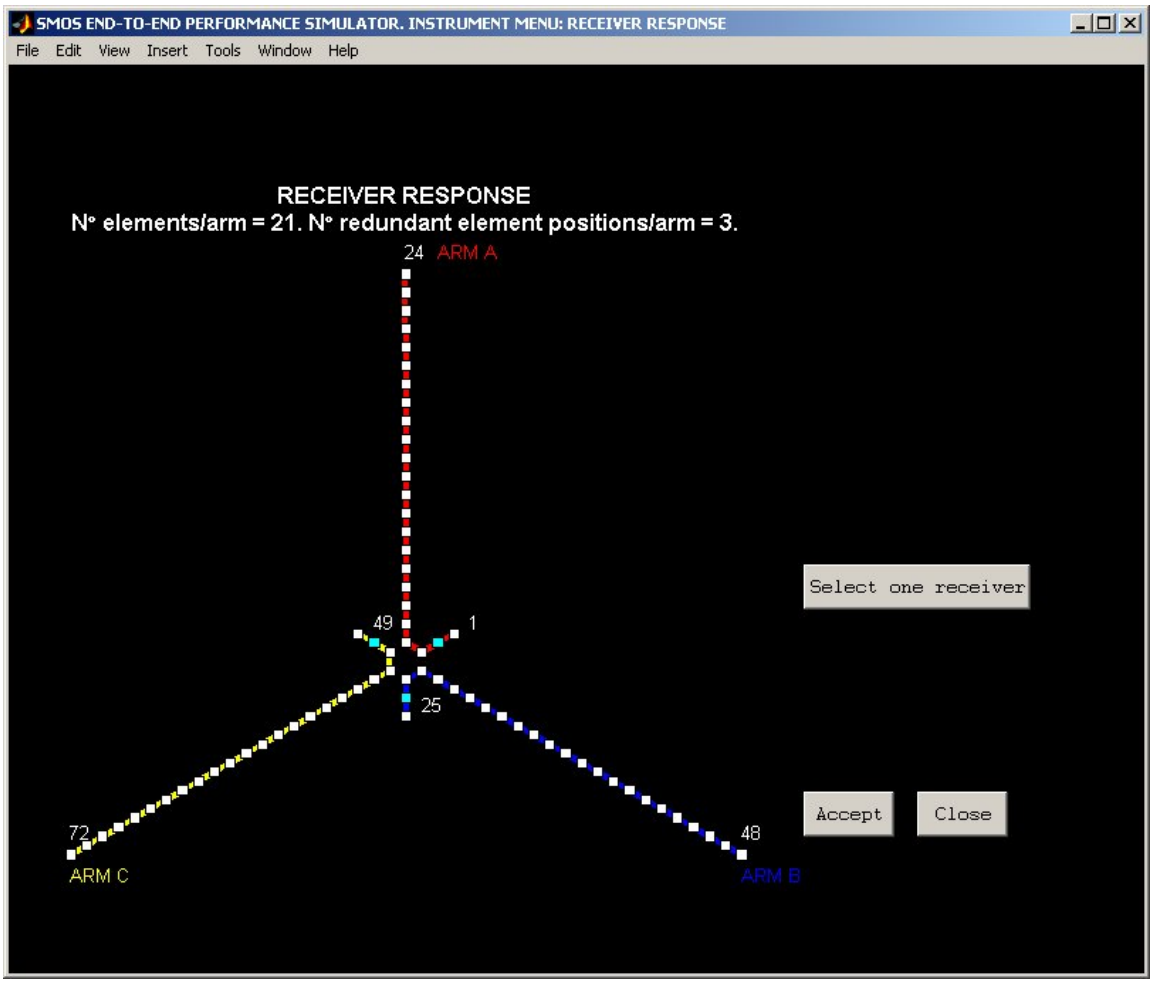


Figure VII.15. Receiver Response *interface, called through the* Instrument FOM*. The user must select the receiver whose main characteristics are to be plotted.*

This interface shows the array as it is currently configured through the *Antenna Characterization* and *Failure of Components* interfaces of the *Instrument Definition* section, in the *Scenario Definition* box. That means, taking into account the steering angle orientation, the number of antennas per arm and the number of redundant antennas per arm, and also the receivers that are failing. The white rectangles represent the receivers, while the rectangles in cyan color represent the ones which are failing, that cannot be selected. Also the numbers of the receivers in the arms' extremes –first redundant and last normal antenna- are shown, in order to help the user to have a visual idea of the array's orientation.

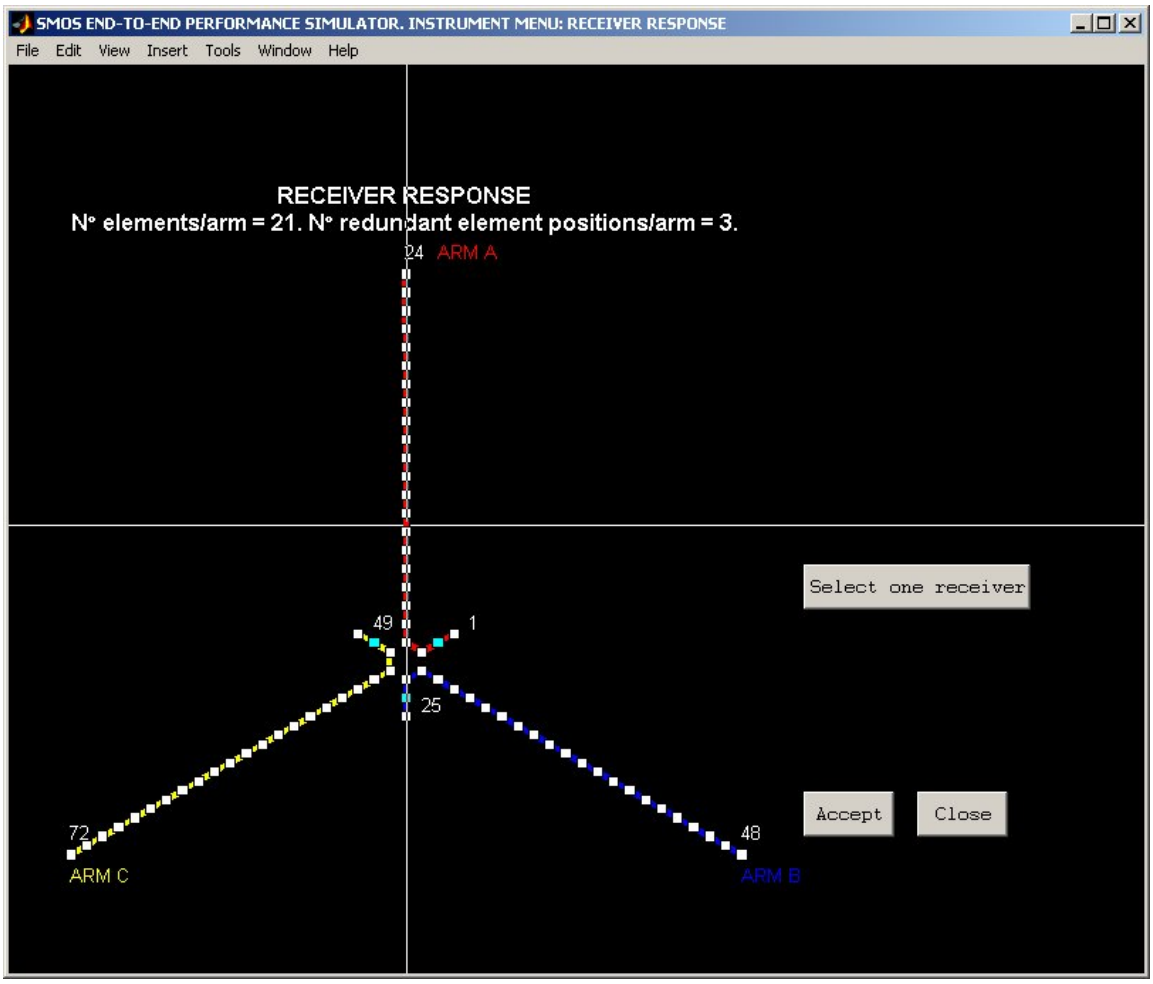


Figure VII.16. Pushing Select one receiver *button, a cursor becomes available to let the user select the suitable receiver. Receivers represented by rectangles in cyan colour cannot be selected.*

The ***Select One Receiver*** button must be pushed to choose the receiver whose main characteristics evolution is requested. Once the referred button is pushed, as seen in Figure VII.16 a cursor becomes available to let the user select the suitable receiver by clicking the left mouse button over one of the white rectangles representing the receivers. The user may notice that rectangles in cyan cannot be selected, as they have been forced to fail through the *Failure of Components* interface in the *Instrument Definition* section. Once the receiver's identifying number is obtained, the user must press the ***Accept*** push button to plot the main characteristics of the selected receiver along the simulation, as it is seen in Figure VII.17.

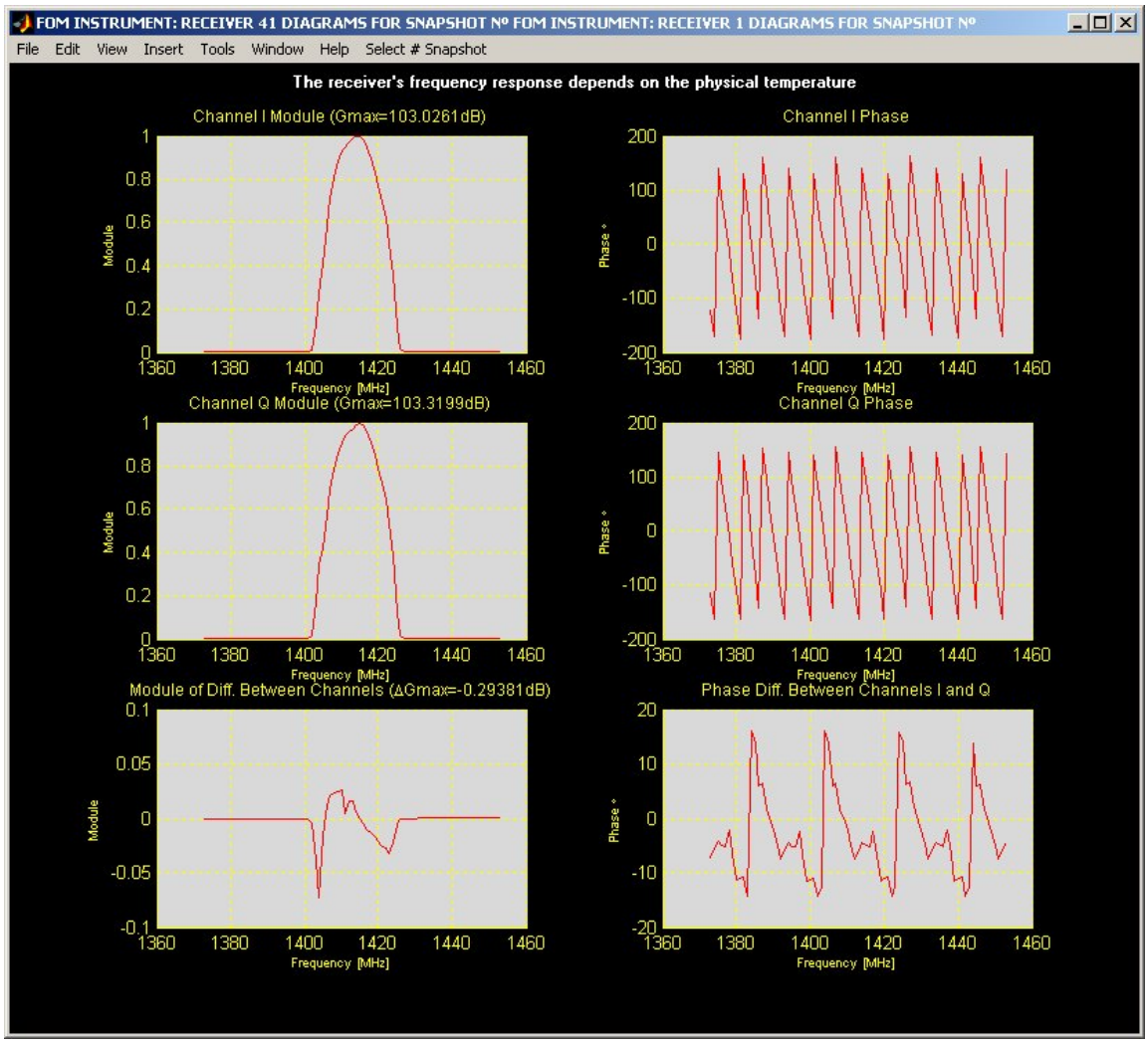


Figure VII.17. Receiver Response *results figure.*

The figure displays six plots: the one in the upper left corner displays the I channel module of the normalized frequency response for the selected receiver and the current selected snapshot; the upper right corner plot shows its I channel phase frequency response; the middle left plot displays the Q channel frequency response normalized module; the middle right plot shows the Q channel frequency response phase; finally, the lower plots display the difference between the normalized modules and phases of the I and Q channels frequency responses. The user may note that the titles of the images containing the module plots display also the value of the maximum gain of the receiver's frequency response in dB[1].

As a receiver frequency response depends on system temperatures, the selected receiver's frequency response for each snapshot of the simulation is plotted in a different figure. As default only the figure for the first snapshot is drawn, but the user can select another snapshot through the usage of the ***Select Snapshot*** # button, located in the top menu bar of the figure, as it can be seen in Figure VII.18.

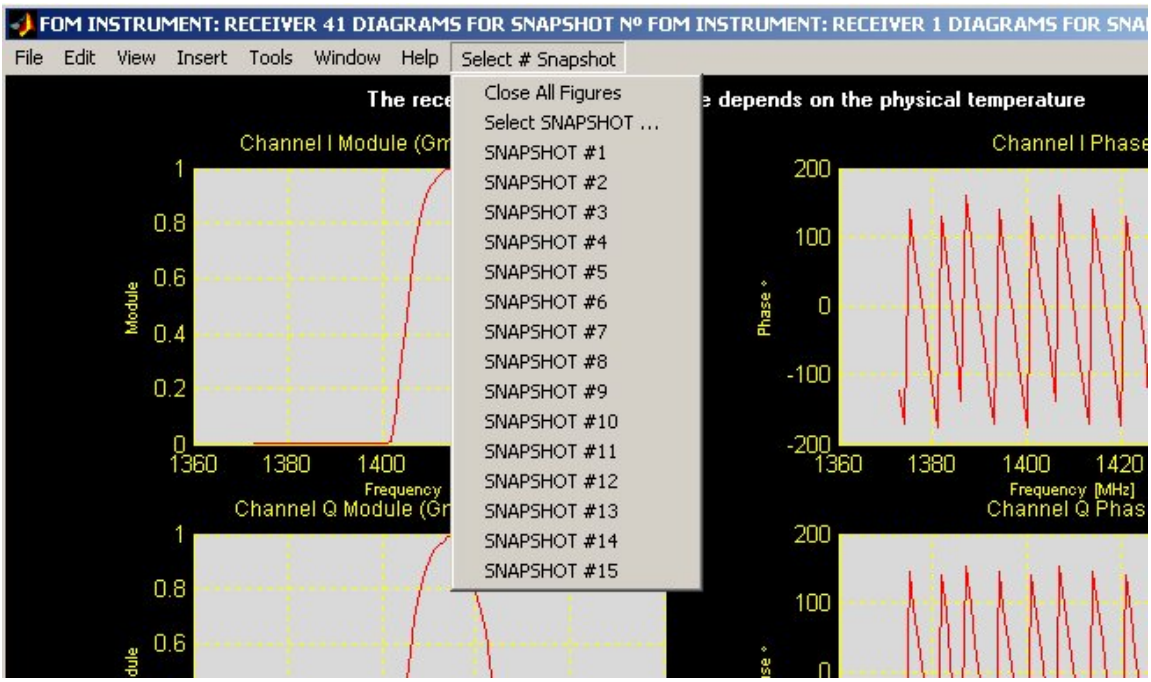


Figure VII.18. The receiver's frequency response plots for another snapshot of the simulation can be drawn using the Select Snapshot # *button in the top menu bar of the figure.*

The ***Close All Figures*** submenu closes all the figures opened through the usage of this tool.

The ***SNAPSHOT #N*** submenu (being N the number corresponding to one of the snapshot stored in the file) leads to displaying the frequency response for the N snapshot, while clicking ***Select Snapshot …*** submenu displays a small window to manage the snapshots to be displayed (see Figure VII.19). This window consists of a list-box, which contains all the snapshots available, and two buttons: ***Apply*** to display the snapshot image selected in the list-box, and ***Close*** to close the window.

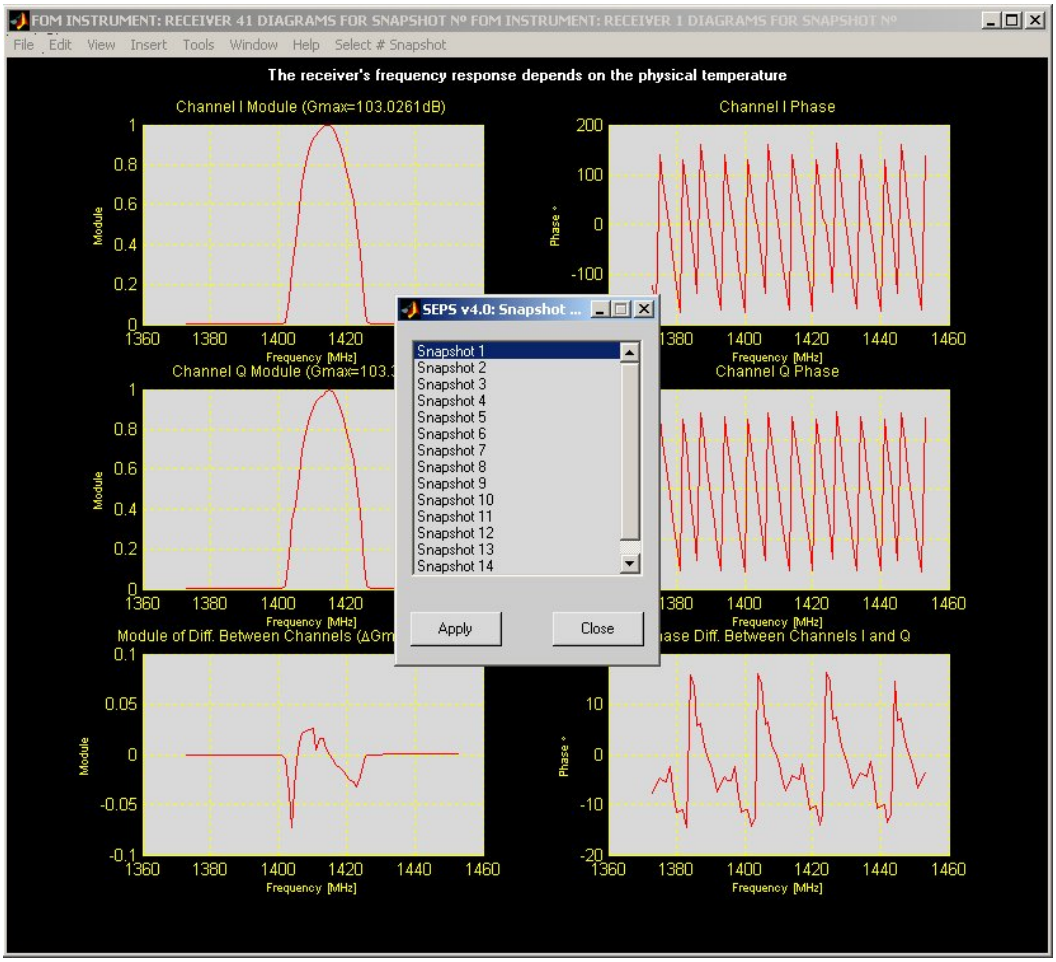


Figure VII.19. Receiver Response *snapshot selection.*

[1] Since receivers' frequency responses were measured in the 1404-1423 MHz range, a periodic extension of the phase has been used, which clearly appears in figure VII.19 right bottom.

VII.2.4. System Temperatures

By clicking the ***System Temperatures*** button the user opens a figure where the antenna temperatures, the physical and noise temperatures of the receivers and other system temperatures are plotted for a user-selected instrument and a snapshot of the simulation. When the referred button is pressed in the *Instrument* menu, the plot shown in Figure VII.20 is presented.

The figure displays six plots: the one in the upper left corner displays the antenna temperatures for the Y-antenna polarization; the upper right corner plot shows the antenna temperatures for the X-antenna polarization; the middle left plot displays the physical temperature of receivers; the middle right plot shows the noise temperature of receivers; the lower left plot displays the antenna temperature measured by the Noise Injection Radiometer; and the lower right plot displays the average of antenna temperatures.

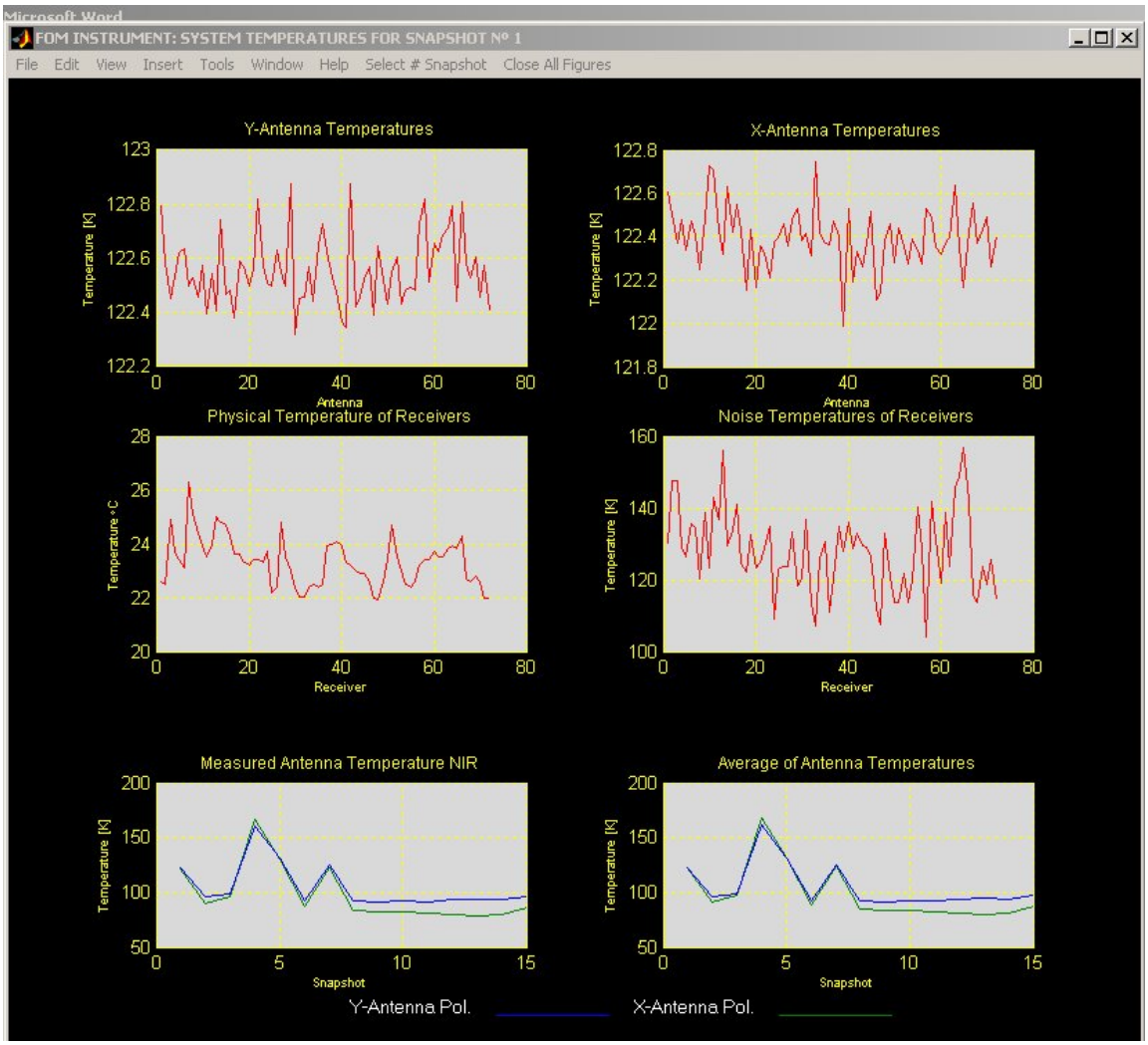


Figure VII.20. System Temperatures *results figure.*

These two lower plots display two curves each one for both Y-antenna and X-antenna polarizations - identified by a legend at the bottom of the figure-, showing the evolution of the temperatures along all the snapshots of the simulation. However the four upper plots in SEPS v4.0 draw only one line corresponding to a snapshot of the simulation. Therefore, the user may press the ***Select Snapshot*** # button located in the top bar menu of the figure to show the temperature curves for another snapshot, as it can be seen in Figure VII.21.

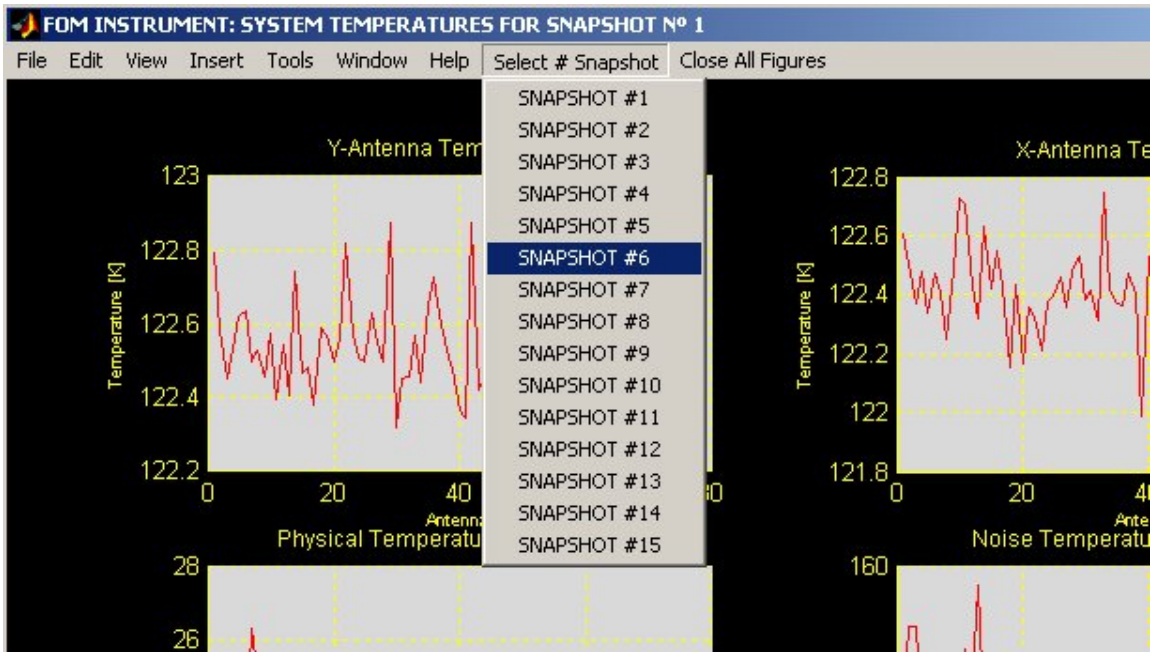


Figure VII.21. The temperature plots for other snapshots of the simulation can be plotted using the Select Snapshot # *group of labels in the top menu bar of the figure.*

The ***Close All Figures*** button closes all the figures opened through the usage of this tool.

VII.2.5. Visibilities

When the user pushes the ***Visibilities*** button two figures are opened, containing single 3D plots each one representing the calibrated visibilities for both Y-antenna and X-antenna polarizations. Figure VII.22 shows the visibilities for the Y-antenna polarization. The user may notice that the value for the (0,0) sample –which represents the antenna temperature- is not plotted in the image, in order to enhance the visualization of all the other samples. The value for the antenna temperature is represented in the text line below the image.

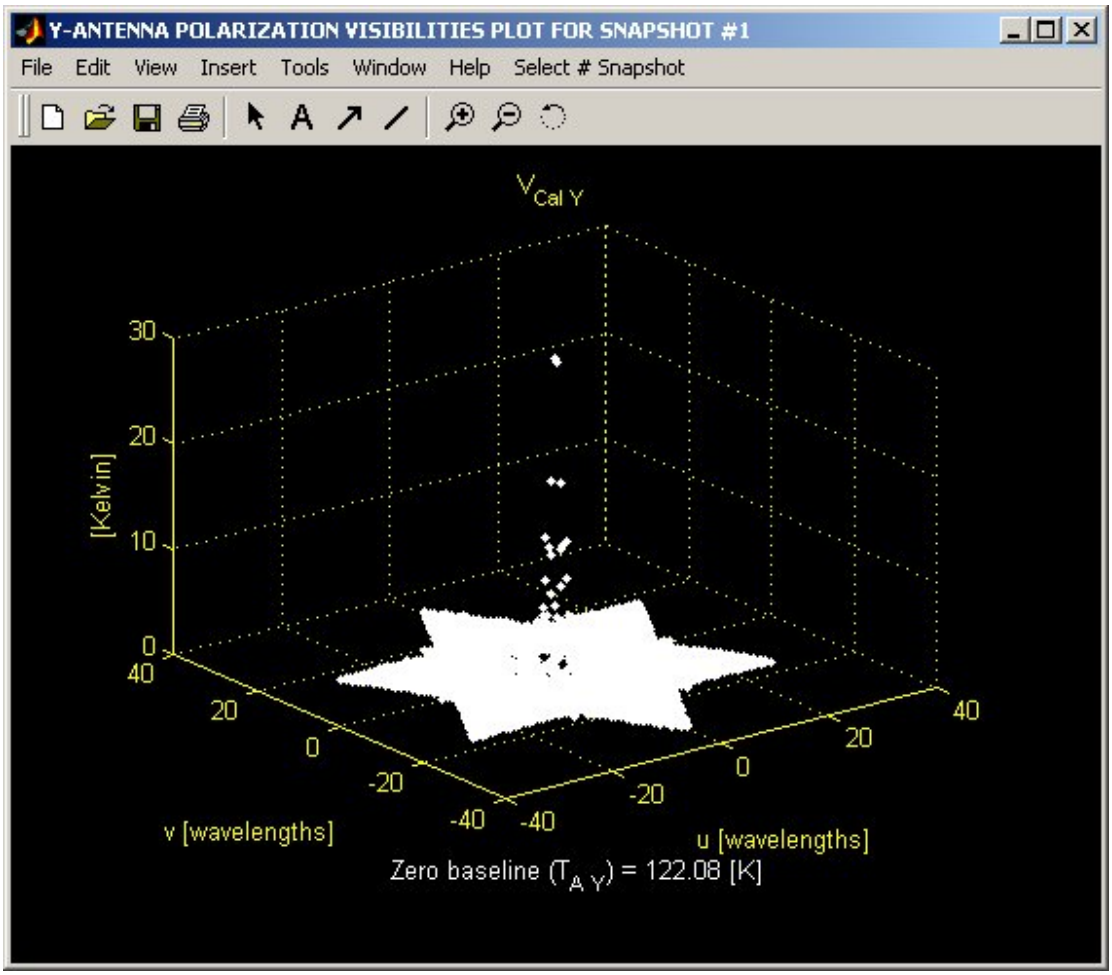


Figure VII.22. Visibilities *figure for the Y-antenna polarization.*

By default, the figures for the first snapshot are shown, but the user can plot the figures for the other snapshots clicking the ***Select # Snapshot*** text label in the upper menu of the window, as it is seen in Figure VII.23.

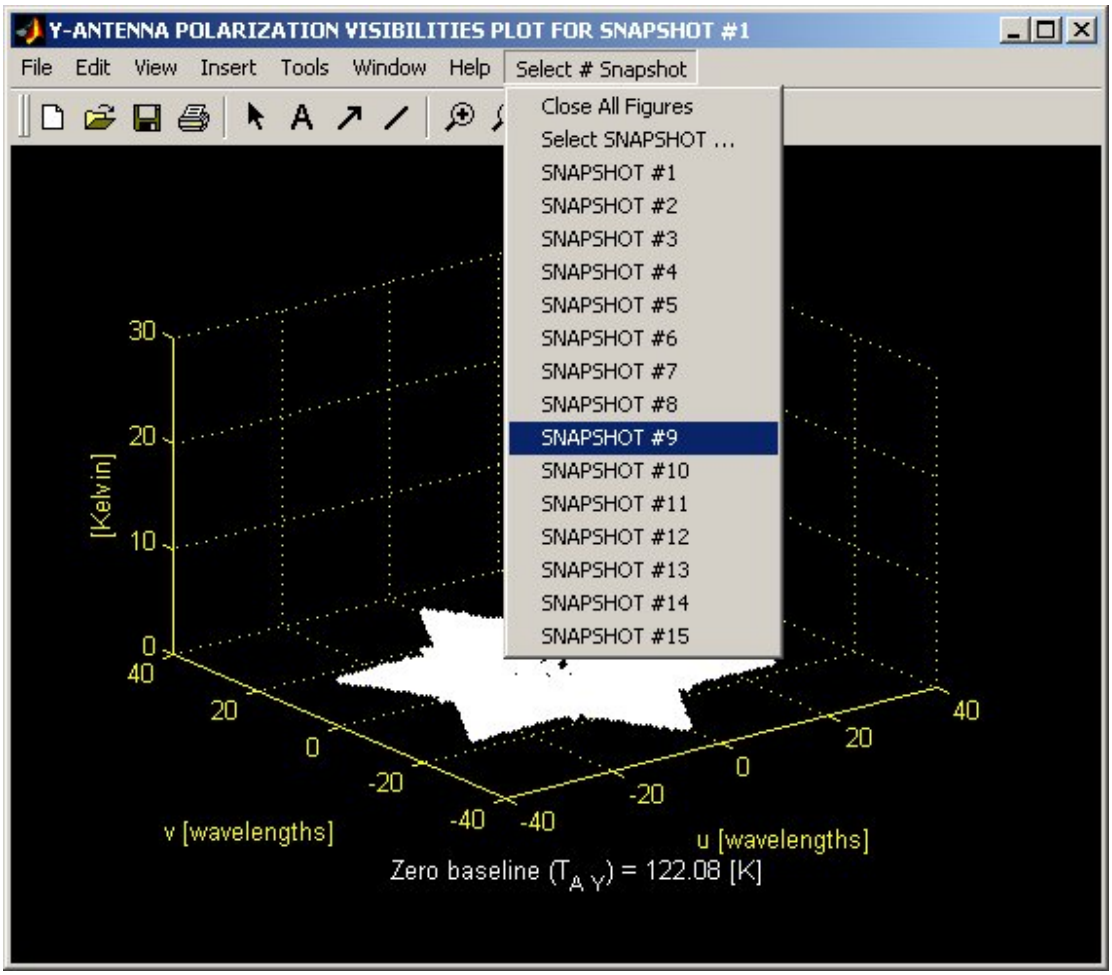


Figure VII.23. Clicking Select # Snapshot *labels, other snapshots visibilities stars can be plotted.*

The ***SNAPSHOT #N*** submenu (being N the number corresponding to one of the snapshot stored in the file) leads to displaying the frequency response for the N snapshot, while clicking ***Select Snapshot …*** submenu displays a small window to manage the snapshots to be displayed (see Figure VII.24). This window consists of a list-box, which contains all the snapshots available, and two buttons: ***Apply*** to display the snapshot image selected in the list-box, and ***Close*** to close the window.

Once the user has finished evaluating the visibilities, the ***Close All Figures*** text label in the upper part of the figure may be used to terminate this FOM Simulation Postprocessing tool.

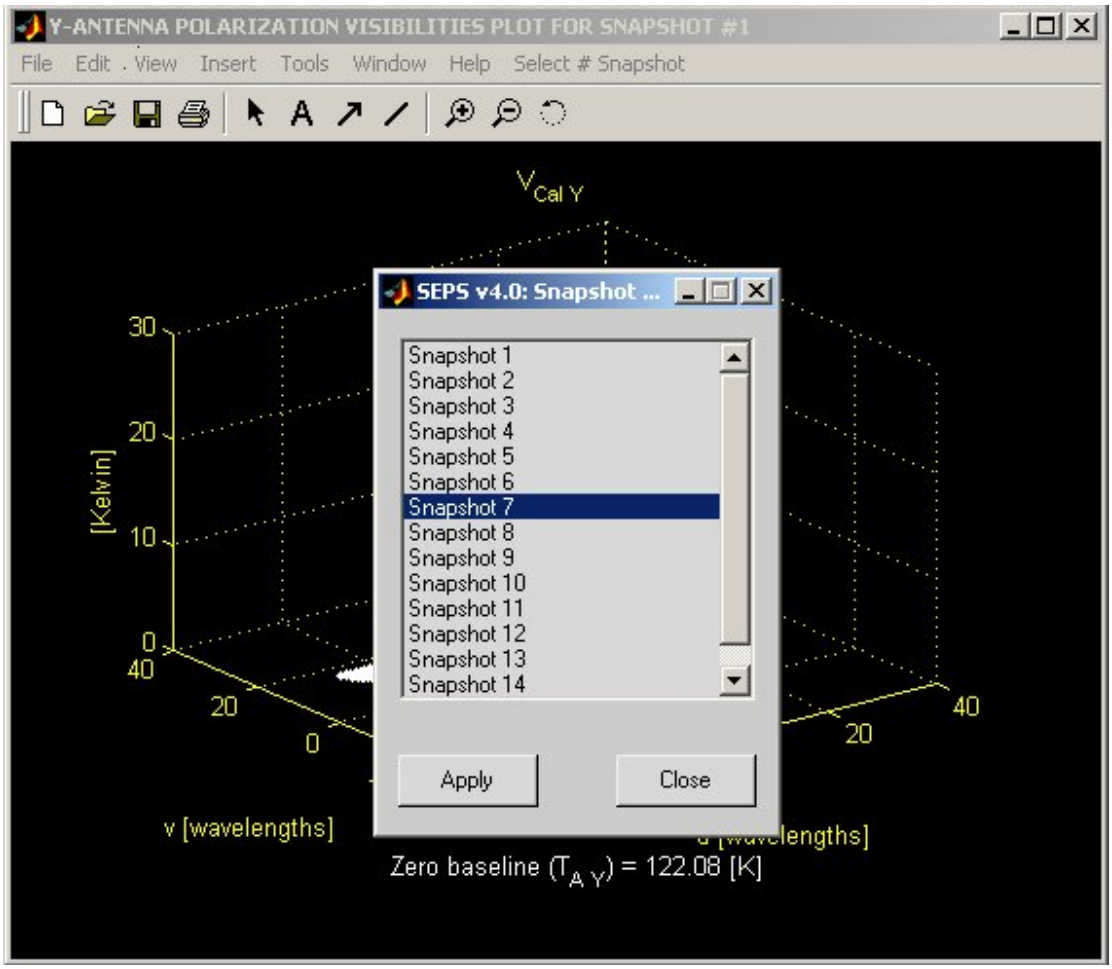


Figure VII.24. Clicking Select SNAPSHOT … *labels, other snapshots visibilities stars can be plotted.*

VII.3. Brightness Temperature Maps

The third set of tools of the *FOM Simulation Postprocessing* section is the *Brightness Temperatures* one. This section allows the user to visualize the resulting simulated data through a comfortable tool where different domains and projections are available. To use this tool the user may click the mouse button on the ***Brightness Temperature*** label in the *FOM Simulation Post-processing* section of the SEPS environment. Then the user must push the ***Reconstructed brightness temperatures*** button to access the *Brightness Temperature Maps* interface, where the different brightness temperature image representation methods are available so that the user can choose the appropriate one. This interface is presented in Figure VII.25.

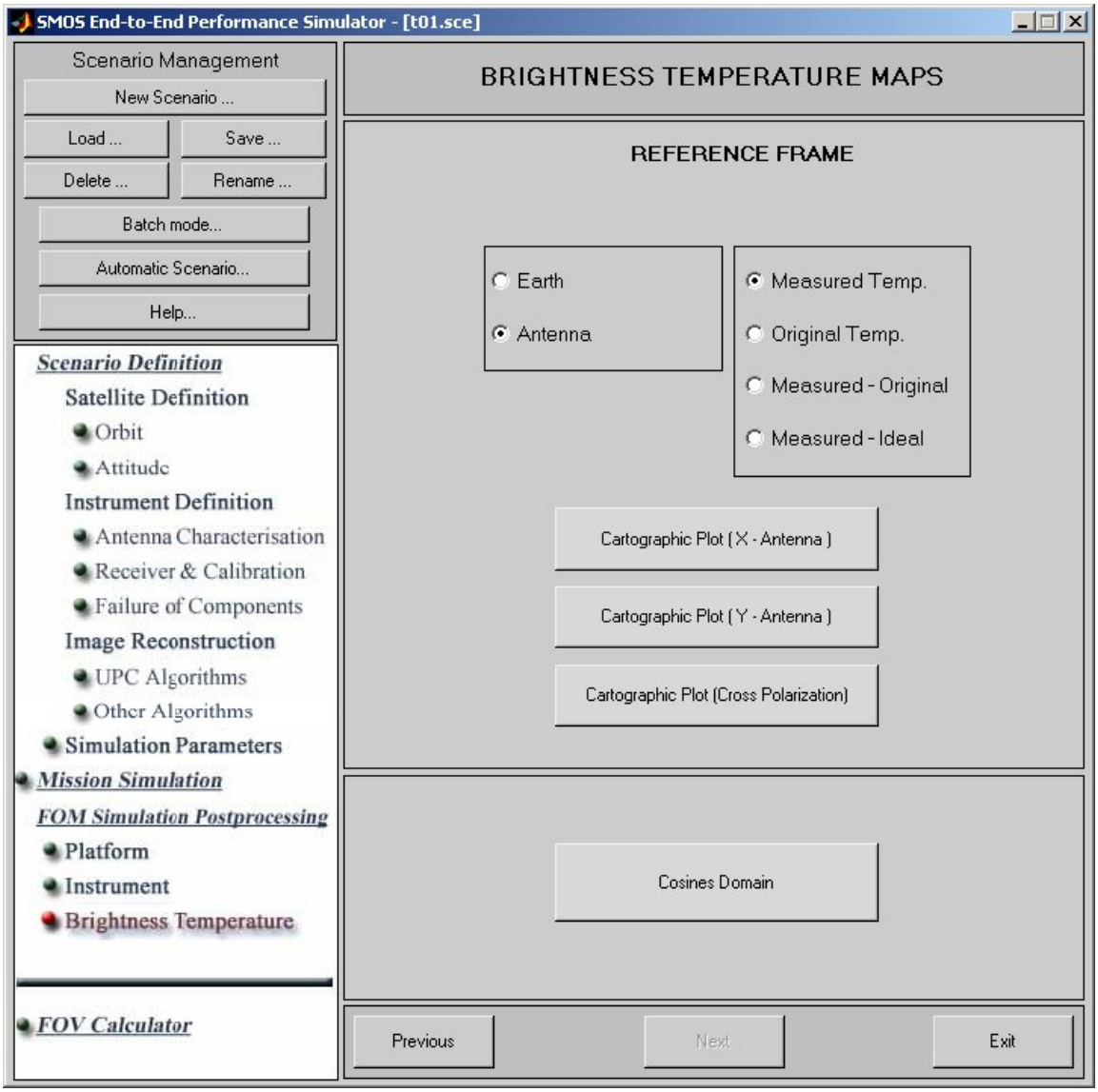


Figure VII.25. Clicking on Brightness Temperature *label leads to display the above Brightness Temperature Maps interface.*

Both frames in the upper part of the interface contain a set of options to select the reference frame where the brightness temperature images are to be plotted. The left square allows the user to select the reference frame where the plot is to be done, the Antenna reference frame or the Earth reference frame. For both reference frames, the second square presents three options:

- to plot the original brightness temperatures as generated at the beginning of the simulation, as outputs of the Brightness Temperature Generator module;
- to plot the brightness temperatures measured using the radiometer and applying the selected image reconstruction algorithm;
- and to plot the error between the original and the measured brightness temperatures.

When the antenna reference frame is selected, also a fourth option is available to plot the error between the measured brightness temperature image and the one that would be measured by an ideal system applying a FFT.

The three lower push buttons in the *Reference Frame* interface, ***Cartographic Plot (Horizontal Polarization)*** –the text in parenthesis is ***X-Antenna*** when the antenna reference frame is selected-, ***Cartographic Plot (Vertical Polarization)*** –***Y-antenna*** if the antenna reference frame is selected-, and ***Cartographic Plot (Cross Polarization)*** -which is only enabled when the antenna reference frame is selected, and only when the full-polarization mode has been used to perform the simulation- present an interface where the user can choose the projection or development to plot the map (that interface is presented in section VII.3.2.1).

The lower push button in this screen is the **Cosines Domain** button, which can be pushed by the user to plot the resulting simulated images in the director cosines domain, that is, the brightness temperature images as seen by the antenna.

Section VII.3.1 details the options for displaying the simulated brightness temperatures and errors in the director cosines domain. Section VII.3.2 explains how the user can display the simulated brightness temperatures and errors in different cartographic projections and developments, also detailing the provided tools to achieve better data visualization.

VII.3.1. Director Cosines Domain

By clicking on the ***Cosines Domain*** button a sequence of figures presenting the simulated data results at the antenna's reference frame is displayed. First, a plot representing the evolution of the accuracy along the simulation is displayed. Figure VII.26 shows the accuracy plots, consisting of two rows of images:

- the first image of the upper row shows the evolution of the bias between the reconstructed or retrieved image, and the image that would be measured by an ideal noise-free system obtaining measurements through an FFT;
- the second image of the upper row shows the evolution of the accuracy between the reconstructed and the ideal image. This accuracy is computed as the root mean square error between both images within a group of points around the alias-free field of view (see Fig. VIII.17 to see the shape of the alias-free FOV);
- the first image of the lower row shows the evolution of the bias between the original image –the one obtained as the output of the Brightness Temperature Generator module computation- and the reconstructed image;
- the second image in the lower row shows the evolution of the accuracy between the original and the reconstructed images.

All images show these statistics for both vertical and horizontal polarizations –or X and Y-antenna polarizations-. A legend identifying the line colour for each polarization is presented at the right hand side of the figure.

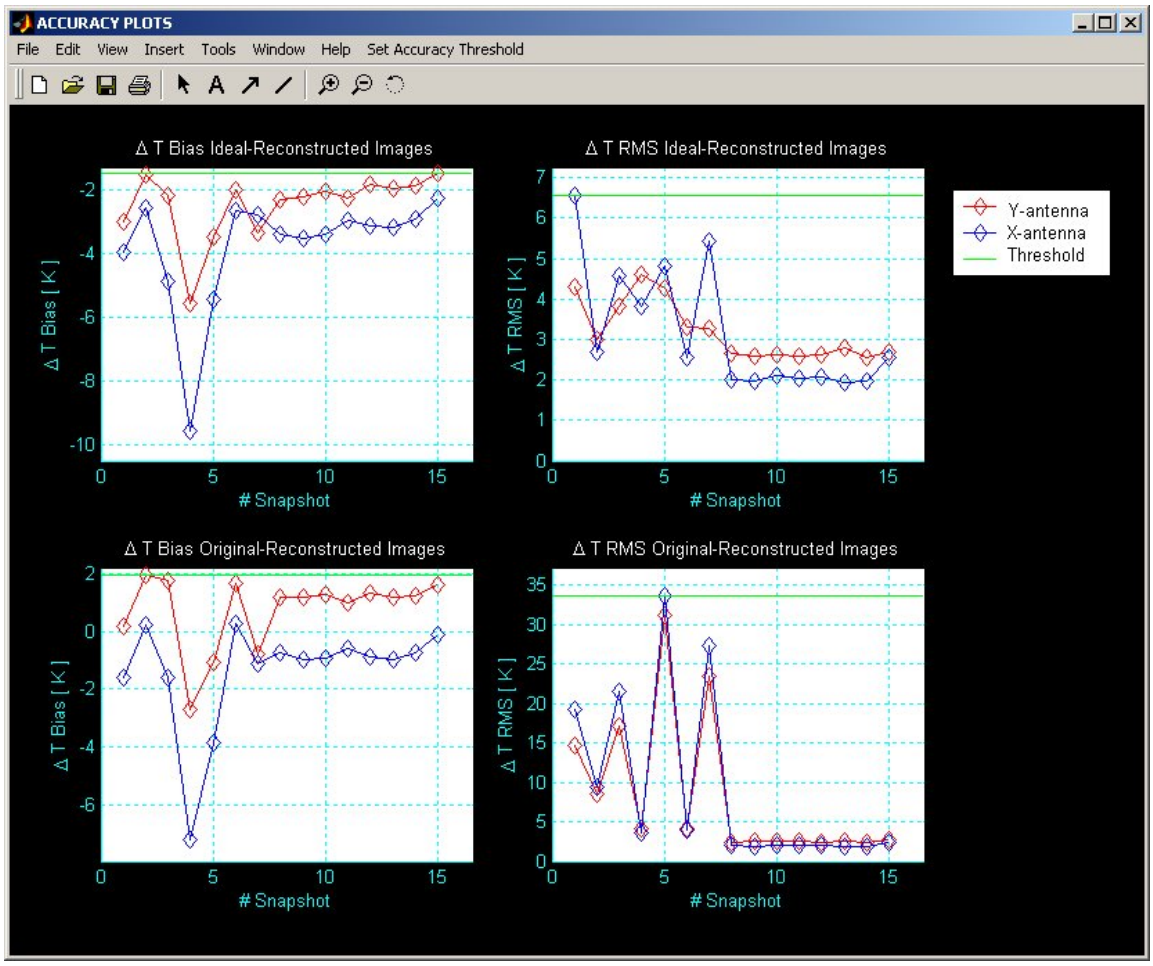


Figure VII.26. Accuracy plots *to show the evolution of the system's performance along the simulation.*

The user may notice that a green line is present in the images, to display thresholds. By default, these thresholds are the maximum values in all snapshots of the simulation, but the user has the possibility of changing them using the ***Set Accuracy Threshold*** tool seen in figure VII.27, in order to visualize better the performance during the simulation. The user may enter the new values for the biases and accuracies thresholds, and then the green line in the corresponding images is then set to the specified value.

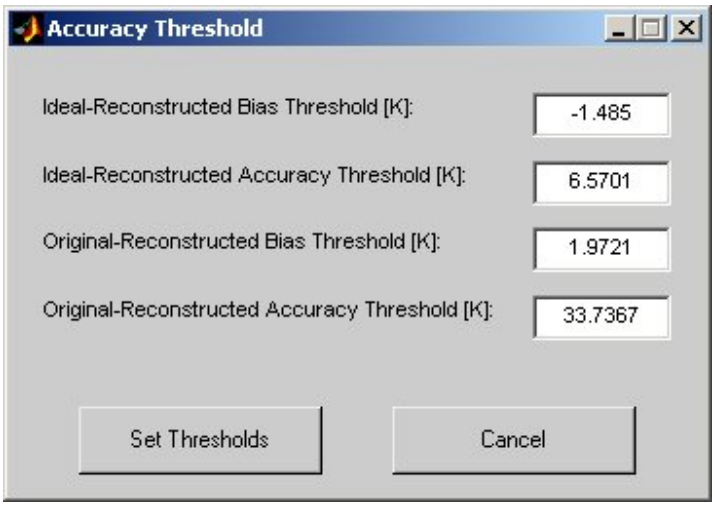


Figure VII.27. Tool for changing the values of the accuracy thresholds

After displaying the *Accuracy Plots* figure, 8 figures are plotted for each snapshot of the simulation. The figures for the first snapshot are displayed by default. These figures plot the generated temperatures as outputs of the *Brightness Temperature Generator* module, the reconstructed brightness temperature images, the ones that would be measured by an ideal system, and also the error images between them

The first window displayed is the one shown in Figure VII.28. The window displays four plots. From left to right and from up to down:

- the upper left plot *Original Temperature* shows the brightness temperature image for the current snapshot as computed by the brightness temperature generator; also the aliases contours and the hexagon are plotted;
- the upper right plot *Original Temperature in Alias-Free FOV* displays the same image, but taking into account only the data inside the SMOS defined field of view; also the hexagon is plotted;
- the lower left plot *FFT-Retrieved Temperature (ideal case)* shows the image that would be measured by an ideal noise-free system applying FFTs;
- and the lower right plot, named as *Retrieved Image*, is the image obtained from the execution of the image reconstruction algorithm.

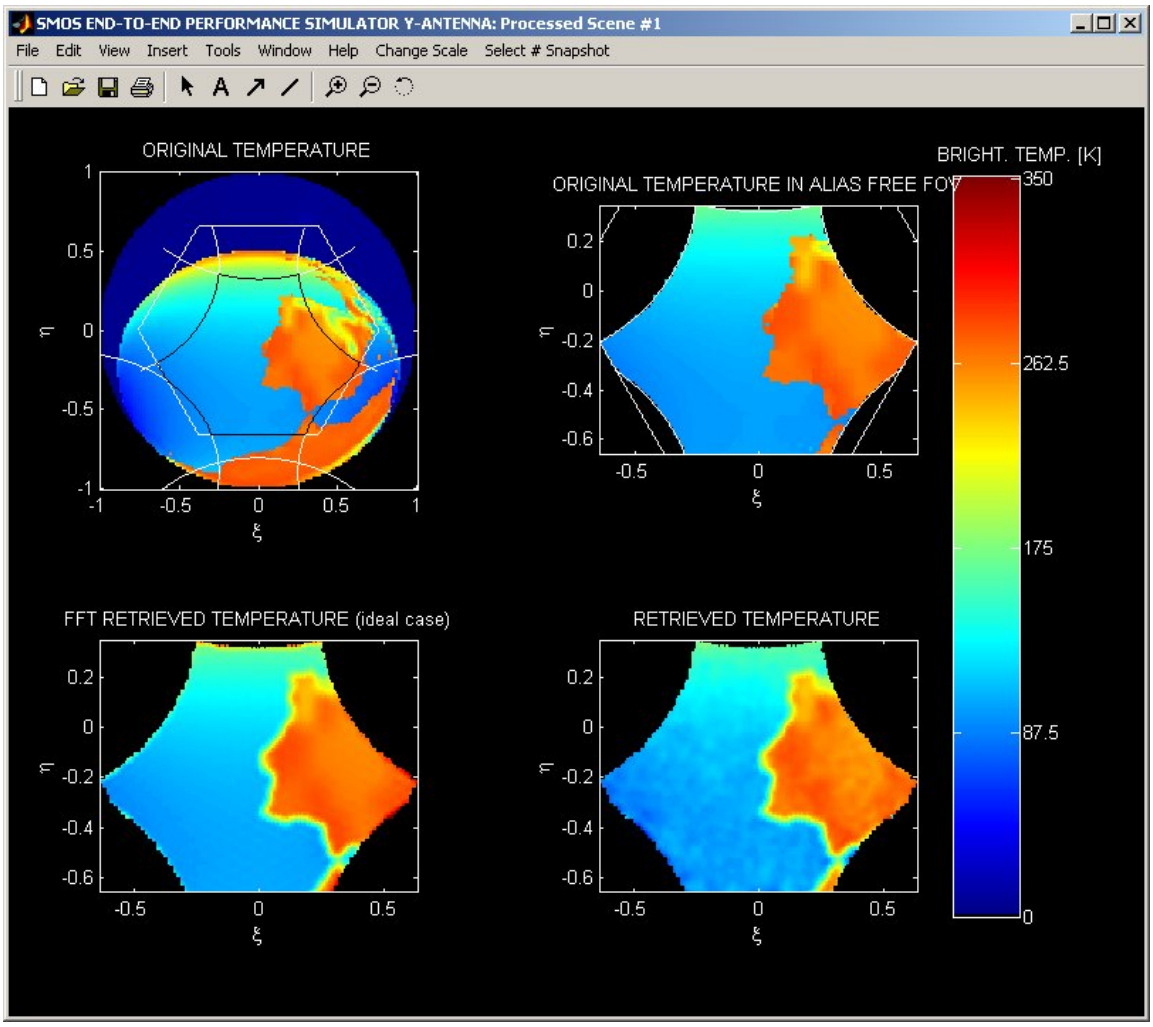


Figure VII.28. Four plots with the original, measured and ideally measured brightness temperatures are shown in this figure

The second figure displays the error images measured as the difference between the retrieved image using the image reconstruction algorithm, and the ideal noise-free FFT-retrieved image –upper image- and the original image as computed by the *Brightness Temperature Generator* module. Figure VII.27 shows an example of this window.

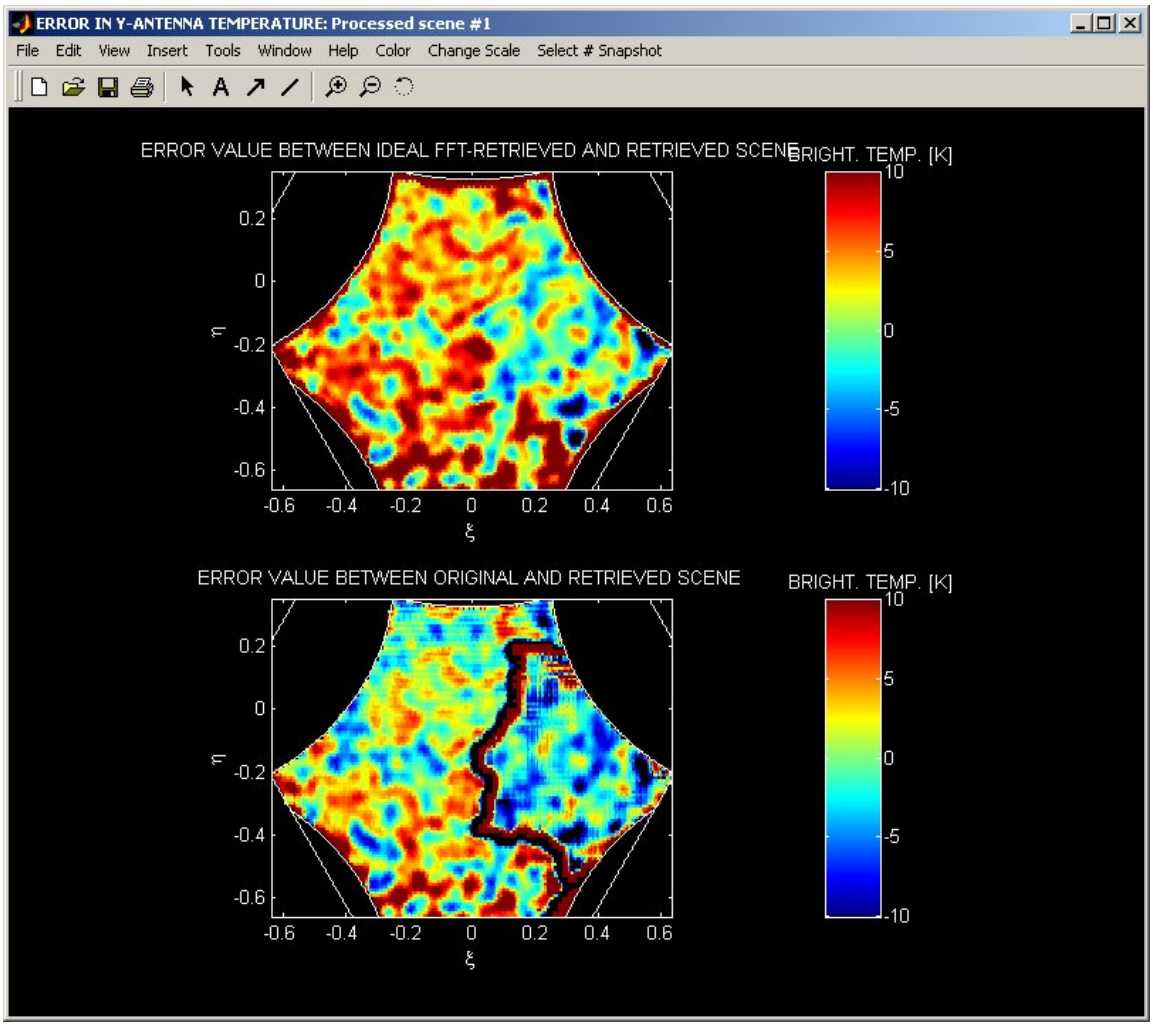


Figure VII.29. Two plots showing the error images between the reconstructed image, and the ideally retrieved and the original images.

These two figures are repeated for the ***Y-antenna*** polarization, the ***Cross (T_{YX}) Polarization brightness temperature*** and the ***first Stokes parameter (I = Tx+ Ty)*** - eight figures altogether, only six if the dual-polarization mode has been used to perform the simulation and hence the cross-polarization figures are not plotted -. Also in the *MATLAB Command Window* a few lines are displayed presenting the bias and the root mean square value of the images computed in the alias-free field of view and in the extended alias-free field of view, for both X-antenna and Y-antenna polarizations.

Y-ANTENNA (Scenario "t30" : Processed Scene #1)
Bias between retrieved and ideal noise-free FFT-recovered TB images at Y-ANTENNA: -2.2049 K inside the alias-free FOV
Bias between retrieved and ideal noise-free FFT-recovered TB images at Y-ANTENNA: -2.7784 K inside extended AF FOV
RMS error between retrieved and ideal noise-free FFT-recovered TB images at Y-ANTENNA: 8.5897 K inside the alias-free FOV

Bias between retrieved and original TB images at Y-ANTENNA: -0.39917 K inside the alias-free FOV
Bias between retrieved and original TB images at Y-ANTENNA: -1.4524 K inside extended AF FOV
RMS error between retrieved and original TB images at Y-ANTENNA: 11.0645 K inside the alias-free FOV

X-ANTENNA (Scenario "t30" : Processed Scene #1)
Bias between retrieved and ideal noise-free FFT-recovered TB images at X-ANTENNA: -6.227 K inside the alias-free FOV
Bias between retrieved and ideal noise-free FFT-recovered TB images at X-ANTENNA: -3.715 K inside extended AF FOV
RMS error between retrieved and ideal noise-free FFT-recovered TB images at X-ANTENNA: 11.4129 K inside the alias-free FOV

Bias between retrieved and original TB images at X-ANTENNA: -5.3807 K inside the alias-free FOV
Bias between retrieved and original TB images at X-ANTENNA: -2.0716 K inside extended AF FOV
RMS error between retrieved and original TB images at X-ANTENNA: 15.7142 K inside the alias-free FOV

YX-ANTENNA (Scenario "t30" : Processed Scene #1)
Bias between retrieved and ideal noise-free FFT-recovered TB images at YX-ANTENNA: 8.458 K inside the alias-free FOV
Bias between retrieved and ideal noise-free FFT-recovered TB images at YX-ANTENNA: 5.5561 K inside extended AF FOV
RMS error between retrieved and ideal noise-free FFT-recovered TB images at YX-ANTENNA: 4.7548 K inside the alias-free FOV

Bias between retrieved and original TB images at YX-ANTENNA: 10.1906 K inside the alias-free FOV
Bias between retrieved and original TB images at YX-ANTENNA: 6.5515 K inside extended AF FOV
RMS error between retrieved and original TB images at YX-ANTENNA: 5.0234 K inside the alias-free FOV

I=TX+TY (Scenario "t30" : Processed Scene #1)
Bias between retrieved and ideal noise-free FFT-recovered TB images at I=TX+TY: -8.4319 K inside the alias-free FOV
Bias between retrieved and ideal noise-free FFT-recovered TB images at I=TX+TY: -6.4934 K inside extended AF FOV
RMS error between retrieved and ideal noise-free FFT-recovered TB images at I=TX+TY: 19.4079 K inside the alias-free FOV

The errors between the deconvolved and the original images are usually large and are concentrated in the high contrast regions, because of the different spatial (angular) resolution of the two images. The errors between the deconvolved and the ideal noise-free FFT-recovered images has a much lower value, in accordance with the results provided by the Radiometric Sensitivity and Radiometric Accuracy FOMs (sections VIII.1 and VIII.2).

The color scale of the brightness temperature images can be changed in order to visualize the resulting images more comfortably. Three possibilities are available for changing the scale of the images after clicking the ***Change Scale*** button:

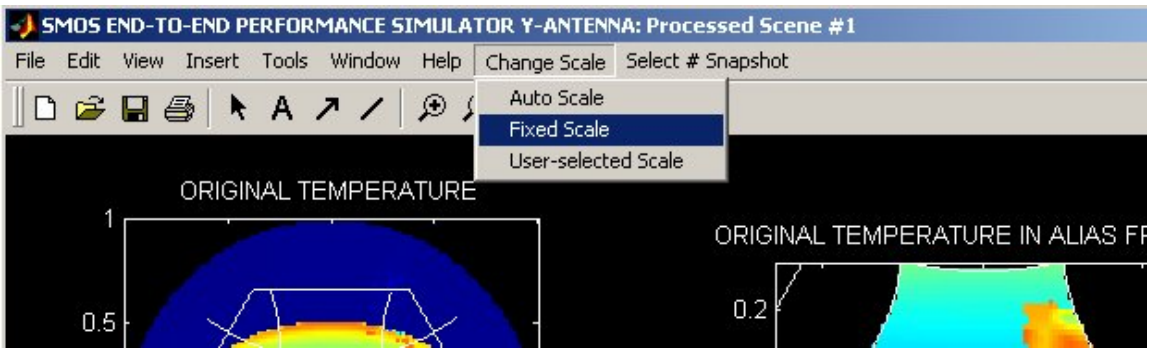


Figure VII.30. Deployable menu displayed when pressing the Change Scale *menu label.*

- ***Auto Scale***: the maximum and minimum values of the images in the figure become the limits of the color scale;
- ***Fixed Scale***: the limits of the scale are fixed, particularly between 0 and 350 K for the absolute value images (Figure VII.28) - between 0 and 700K for the Stokes I parameter figure -, and between -10 and 10 K for the error images (Figure VII.29) – between -20 and 20 K for the Stokes I parameter errors figure -.
- ***User-selected Scale***: the window presented in Figure VII.31 allows the user to supply the required values for the maximum and minimum of the new scale, in the edit boxes.

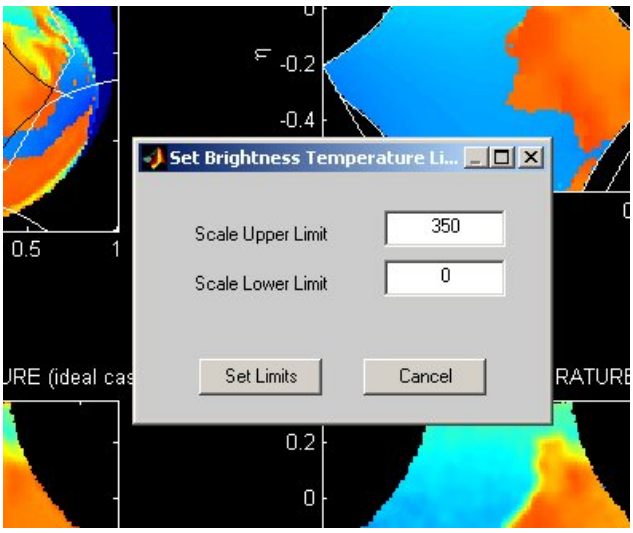


Figure VII.31. The user must supply the maximum and minimum values for the new scale.

Figure VII.32 shows the resulting figure of changing the color scale. In this example, Figure VII.28 has been re-scaled to be within the range [50-400] Kelvin –Fixed Scale-, and the colors of the images change.

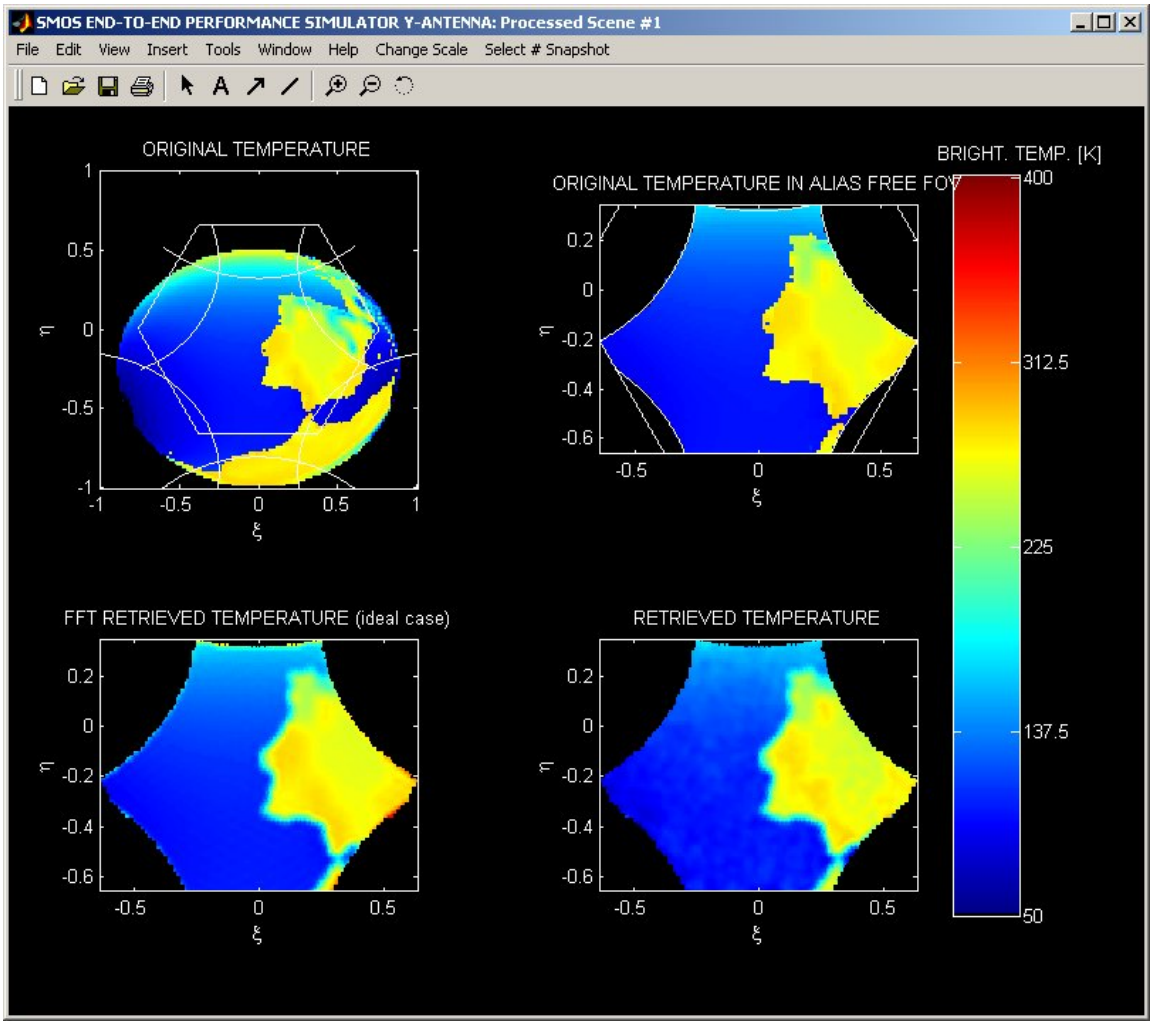


Figure VII.32. Figure VII.30 after changing the scale.

By default, the figures for the first snapshot are shown, but the user can plot the figures for the other snapshots clicking the ***Select # Snapshot*** text label in the upper menu of the window, as it is seen in Figure VII.33.

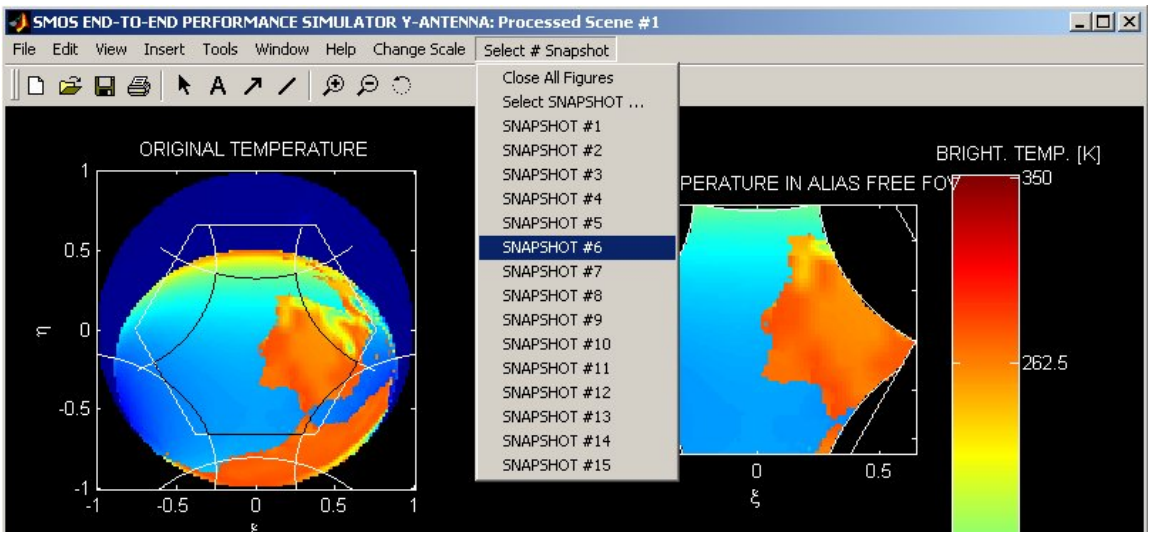


Figure VII.33. Clicking Select # Snapshot *labels, other snapshots cosines domain can be plotted.*

The ***SNAPSHOT #N*** submenu (being N the number corresponding to one of the snapshot stored in the file) leads to displaying the N snapshot, while clicking ***Select Snapshot …*** submenu displays a small window to manage the snapshots to be displayed (see Figure VII.34). This window consists of a list-box, which contains all the snapshots available, and two buttons: ***Apply*** to display the snapshot image selected in the list-box, and ***Close*** to close the window.

Clicking on ***Close All Figures*** to close the cosine domain representation tool and all the previously displayed figures.

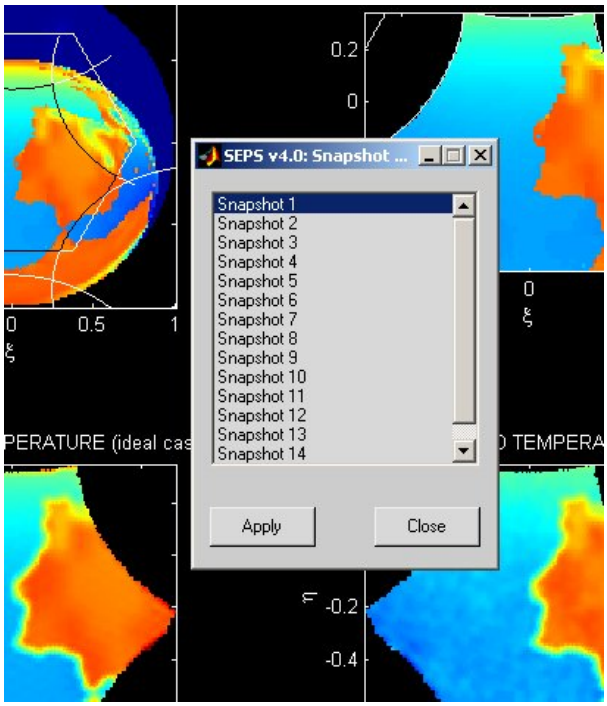


Figure VII.34. Clicking Select SNAPSHOT … *labels, other snapshots can be plotted.*

VII.3.2. Cartography

The *Brightness Temperatures* interface provides the user with the possibility of visualizing the simulated data results in different cartographic projections and developments. The interface shown at the beginning of section VII.3 (Figure VII.25) allows the user to select a data type (choosing the reference frame and the type of simulated brightness temperatures, as explained in the referred section) and launch a Cartographic Representation Method figure, which is the starting point for the entire cartography module.

IMPORTANT ➔ *It may happen that the user is not interested in the cartography module, and therefore considers the installation of The Mathworks, Inc. Mapping Toolbox an unnecessary acquisition. The Mapping Toolbox is intended to be used only in the projection of the simulated parameters on Earth's surface and in the Cartography module. If none of these two features is needed, the user can* ***disable*** *the* **Use cartography module** *check box in the* **Mission Simulation** *interface before the simulation starts.*

Section VII.3.2.1 examines the cartographic tool's functionality. Section VII.3.2.2 details the available projections and developments and overviews their particularities. Finally, section VII.3.2.3 reviews the new functionalities to plot the different simulated data and graphically evaluate their evolution along the simulation.

VII.3.2.1. Functionality

This section overviews the available tools to the user through the cartography figures in order to handle them. Section VII.3.2.1.1 details the functionality of the main cartography interface window created to visualize a certain brightness temperature type of data. The subsequent sections explain the tools present in the figures where the data is projected.

VII.3.2.1.1. Cartographic Representation Method window

Clicking the **Brightness Temperature** label in the FOM Simulation Postprocessing section leads to launch an interface to project the simulated data in a group of selected projections and developments, as it is seen in Figure VIII.35.

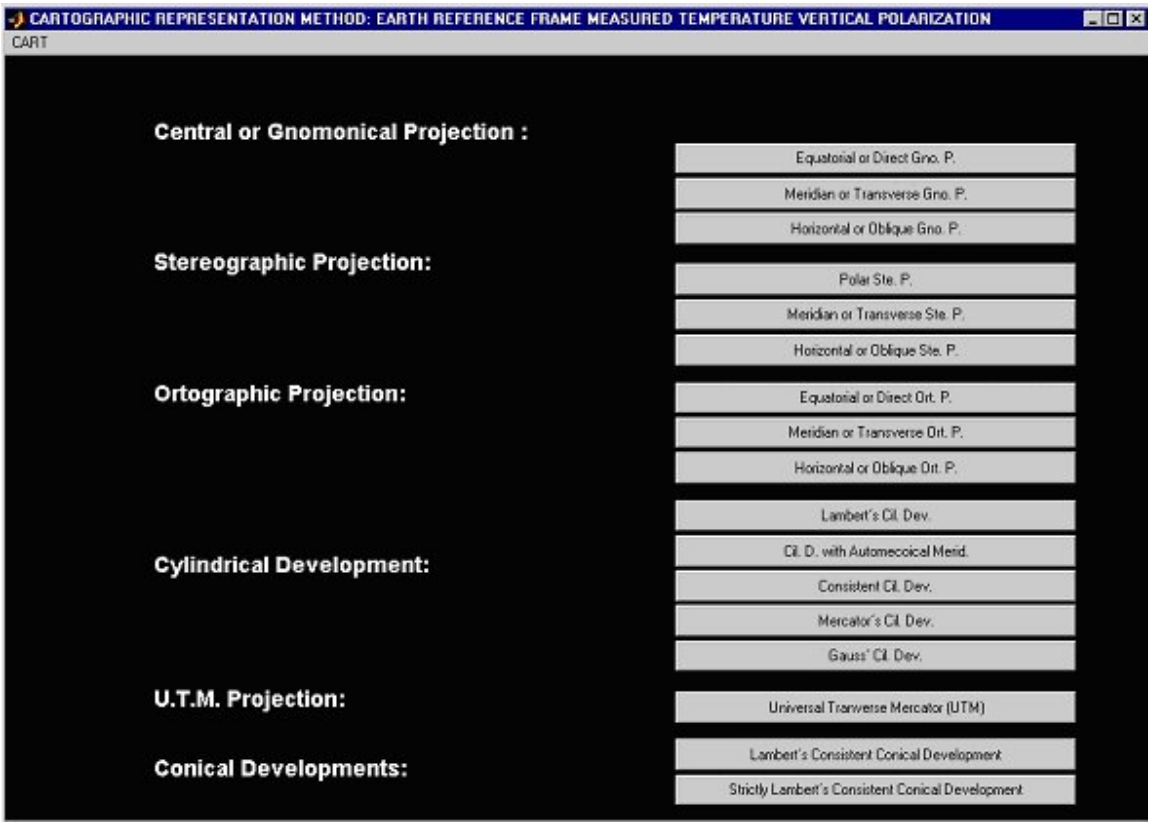


Figure VII.35. This window allows the user to select the projection or development. Clicking the corresponding button launches a figure where the projected map is displayed.

Through the buttons in this window, the user may select the projection for the current type of data specified in the *Brightness Temperature* interface. The user may note that a description of the type of data being plotted through this cartography window is available in the figure's upper strip's character string (in Figure VII.35 example, this string is "Earth reference frame, measured temperature, vertical polarization"). This type of data is the one plotted at user's request, although in SEPS v4.0, the user can change the type of data without having to open another *Cartographic Representation Method* (see section VII.3.2.3 to review these capabilities).

When a projection is selected a new figure is opened where the projection of the current specified type of data is performed. As in the *Cartographic Representation Method* window, the projected type of data and the projection's description can be found in the

figure's upper strip's character string. An example for the *Horizontal Orthographic Projections* is shown in Figure VII.36:

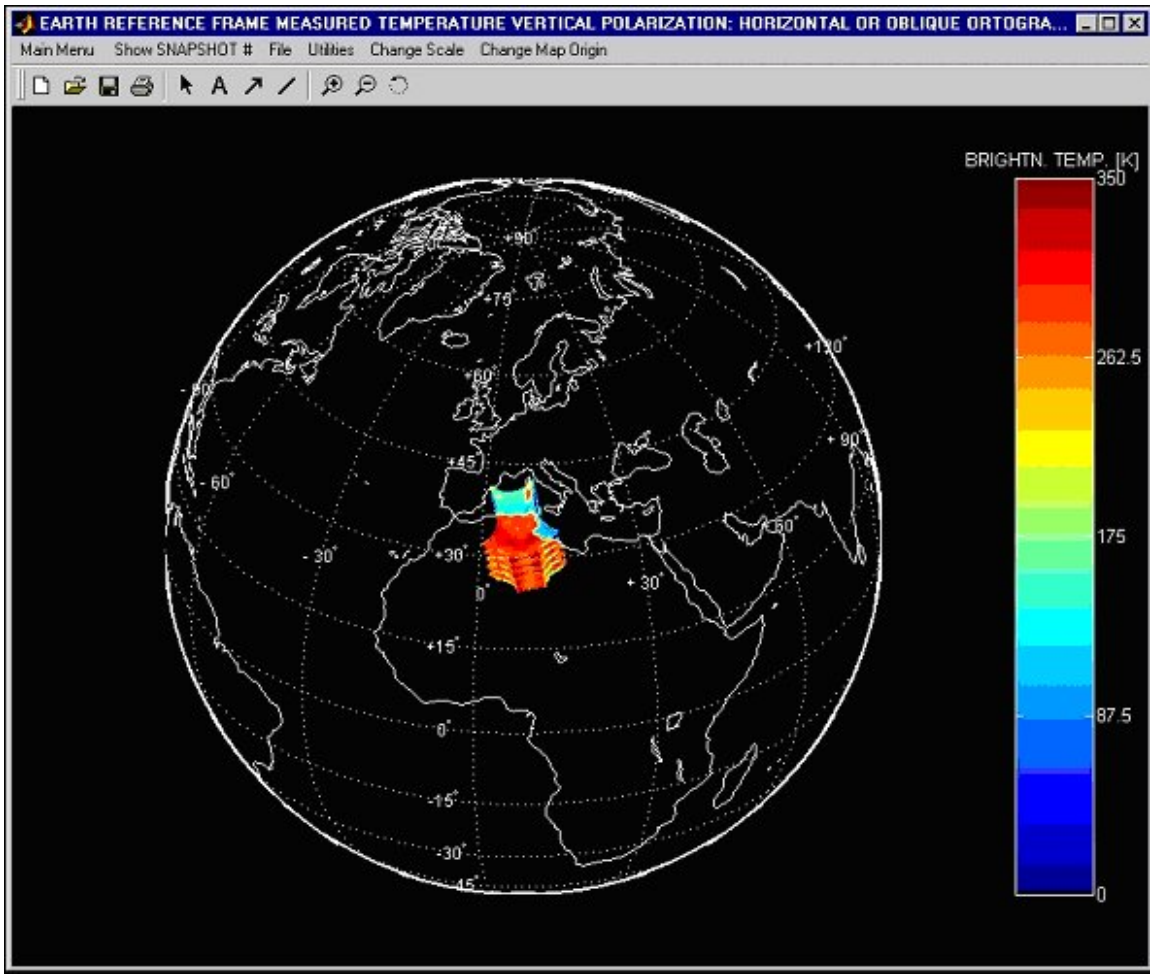


Figure VII.36. Cartographic projection launched from the Cartographic Representation Method window in Figure VII.35.

The following subsections detail the possible operations that the user is allowed to perform when the figure for a selected projection is opened. These actions are requested through the menu labels added to the figure's menu bar –upper grey part of the figure. From now on, the images use always the above example.

To leave the *Cartographic Representation Method* the user must press the ***CART*** label in the left upper corner of the window. The window shown in Figure VII.37 is

displayed, and the user may select ***Yes*** to close the Cartography tool –the Cartographic Representation Method and all its dependant figures- or ***No*** to cancel this operation.

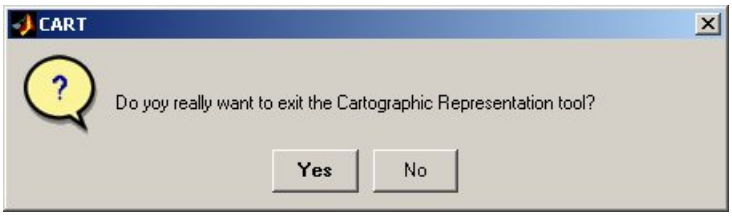


Figure VII.37. Dialog window to confirm that the user wants to exit the Cartographic tool.

VII.3.2.1.2. Main Menu

When the user presses the ***Main Menu*** label, the current projection figure and all its dependent figures (in the case it is a *Universal Transverse Mercator* projection) are closed. Then, the *Cartographic Representation Method* window (see Figure VII.35) corresponding to the type of data plotted in the closing projection figure –specified in the figure's top bar- is forced to be the current window.

VII.3.2.1.3. Show Snapshot #

The ***Show Snapshot #*** menu label allows the visualization of a single snapshot. Depending on the snapshot time value, the pixels of different snapshots can meet the same geographic areas, that is, the upper field-of-view zone of one snapshot can coincide with the lower field-of-view zone of the following snapshot. This establishes the need for a cartographic tool to allow the user analyzing the pixels for the different snapshots separately.

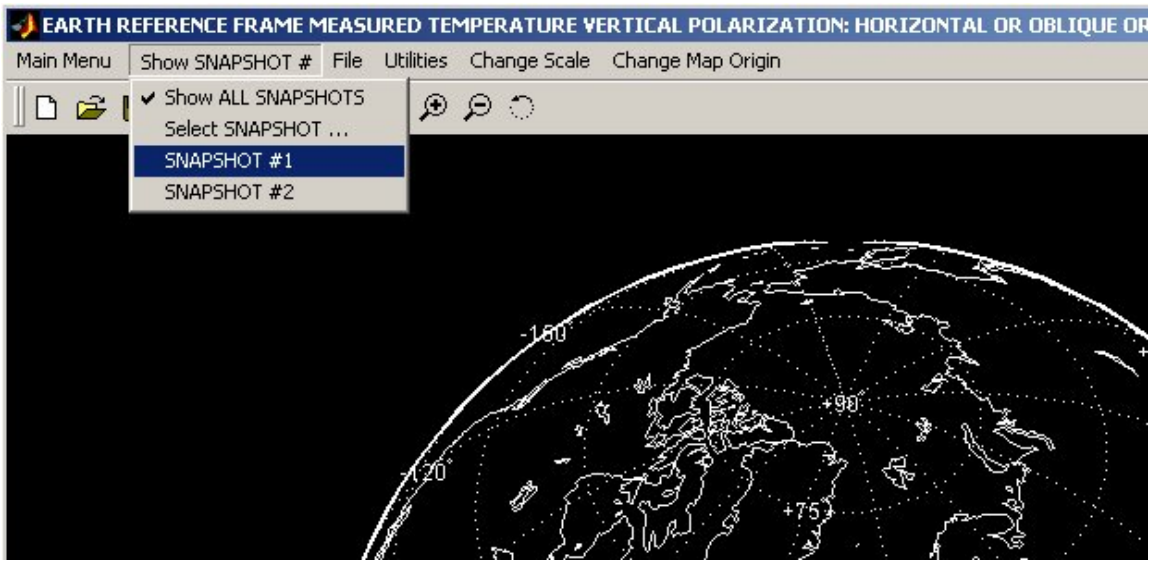


Figure VII.38. The Show Snapshot # label in the figure's menu allows the user visualize one single snapshot or the complete simulation.

Pressing the ***Show Snapshot #*** label a set of sub-labels is expanded. Each one forces the representation of a particular single snapshot; and the final sub-label ***Show All Snapshots*** forces the figure to represent the whole amount of data, that is, the super-imposition of the pixels for adjacent snapshots. If the snapshot time value is lower enough, later snapshot's pixels will appear super-imposed over the preceding ones. The Figure VII.39 shows an image where a single snapshot has been projected.

Clicking ***Select Snapshot ...*** submenu displays a small window to manage the snapshots to be displayed. This window consists of a list-box, which contains all the snapshots available, and two buttons: ***Apply*** to display the snapshot image selected in the list-box, and ***Close*** to close the window.

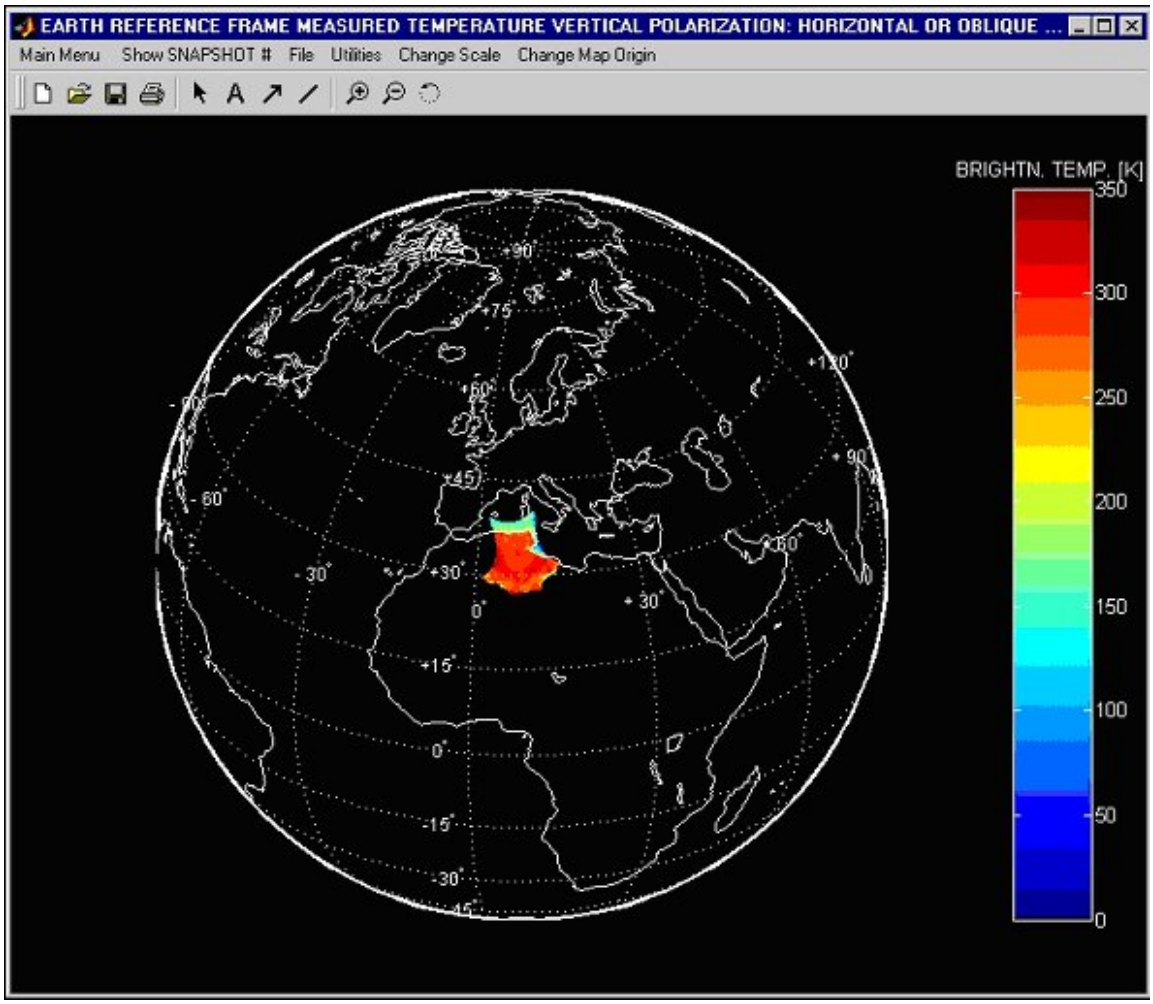


Figure VII.39. One single snapshot displayed in the projection figure.

VII.3.2.1.4. File

The File label makes available for the user some tools for saving / copying the projected figure. Figure VII.40 shows how these tools are called.

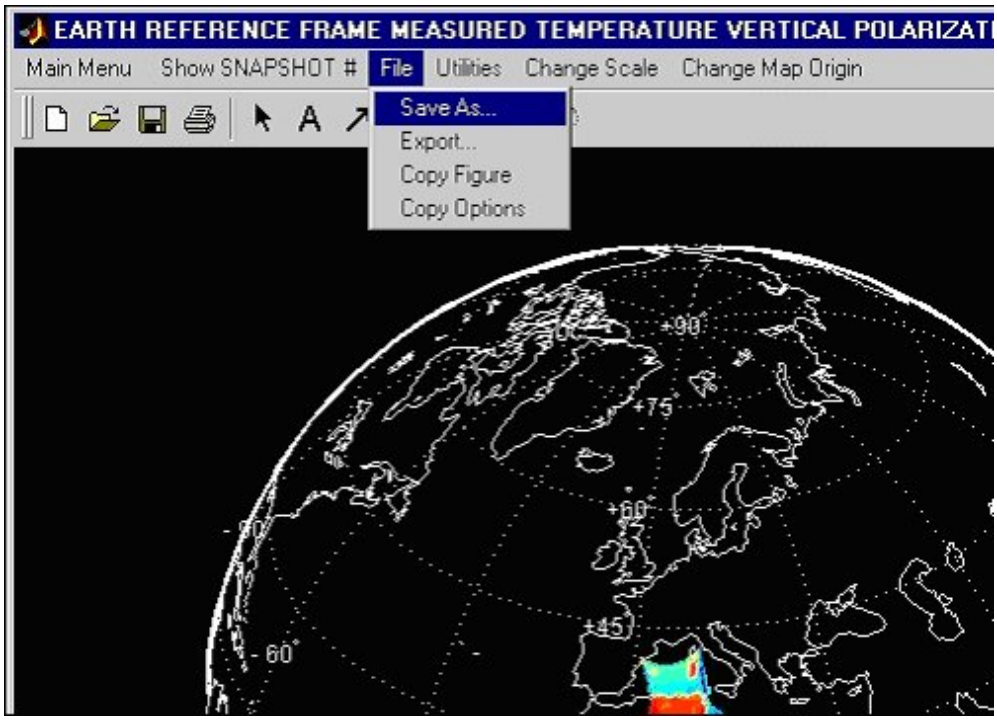


Figure VII.40. Tools for managing the projected figure

The ***Save As*** tool allows the user to store the entire figure and its capabilities in a file with the MATLAB *.fig* extension for figures. This means that the figure's aspect can be edited later, and that some tools that do not need intermediate files to execute can be used later. When the label is pressed, a dialog box is opened to request the user to supply the file's name.

The ***Export*** label launches a dialog box to ask the user the format requested to store the figure. Figure VII.41 shows the dialog box and some of the formats (JPG, BMP, TIFF, etc…). Obviously, the resulting graphical file cannot be edited as it can be done with a MATLAB *.fig* file.

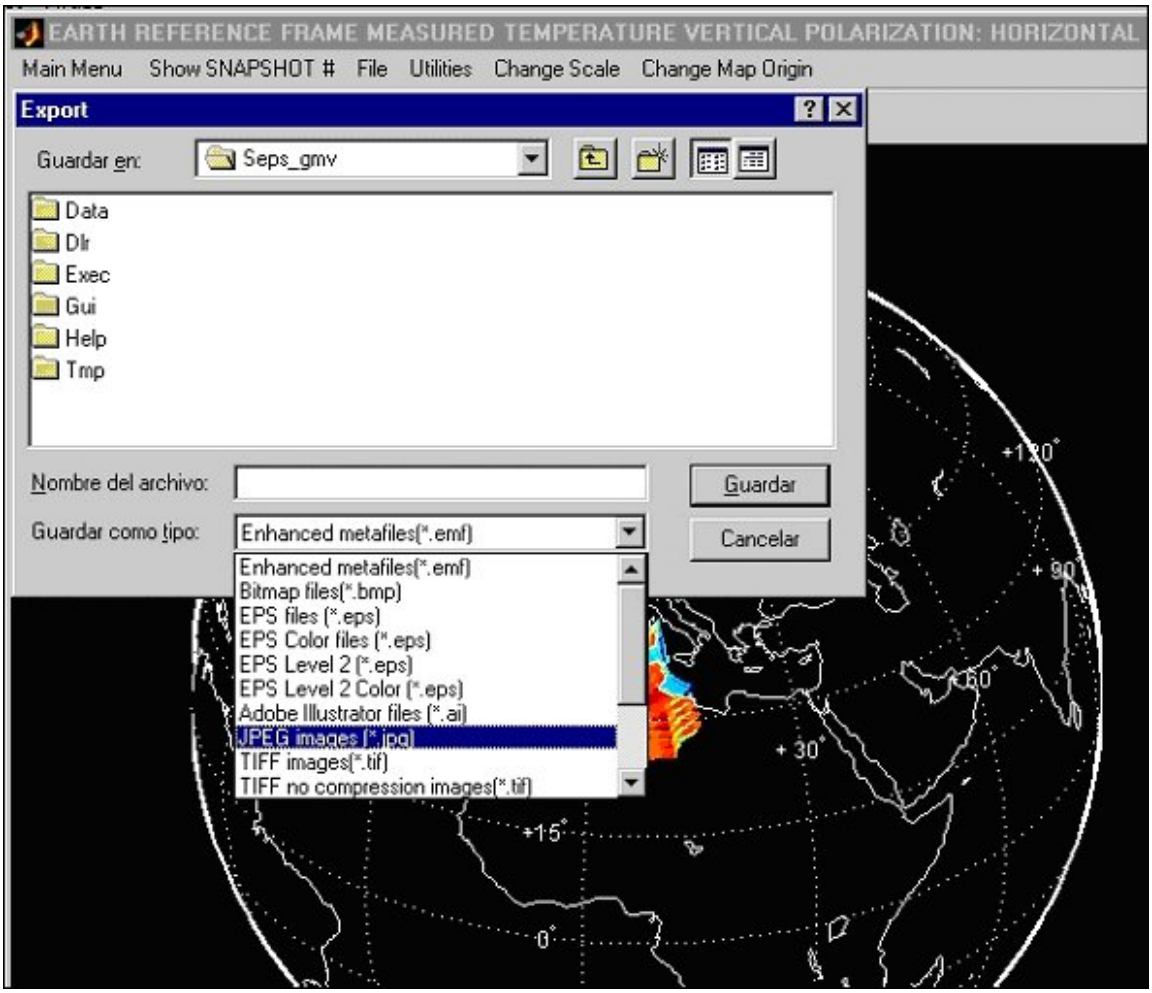


Figure VII.41. The Export tool allow the user to store the figure's appearance indifferent graphical formats

Finally, the ***Copy Figure*** label allows the user to store in memory the figure, following the ***Copy Options*** –see MATLAB documentation for more information on copy options for MATLAB figures-.

VII.3.2.1.5. Cursor Position

Clicking the ***Utilities*** menu label in the figure's menu bar, some tools become available. These tools are *Cursor Position* and *Axes* managing –which is reviewed in he next section. Selecting the ***Cursor Position*** menu label, the cartographic tool provides the user with the possibility to select a projected pixel to get its associated values. When the

label is pressed the user must select one point in the figure by clicking on it with the left mouse button.

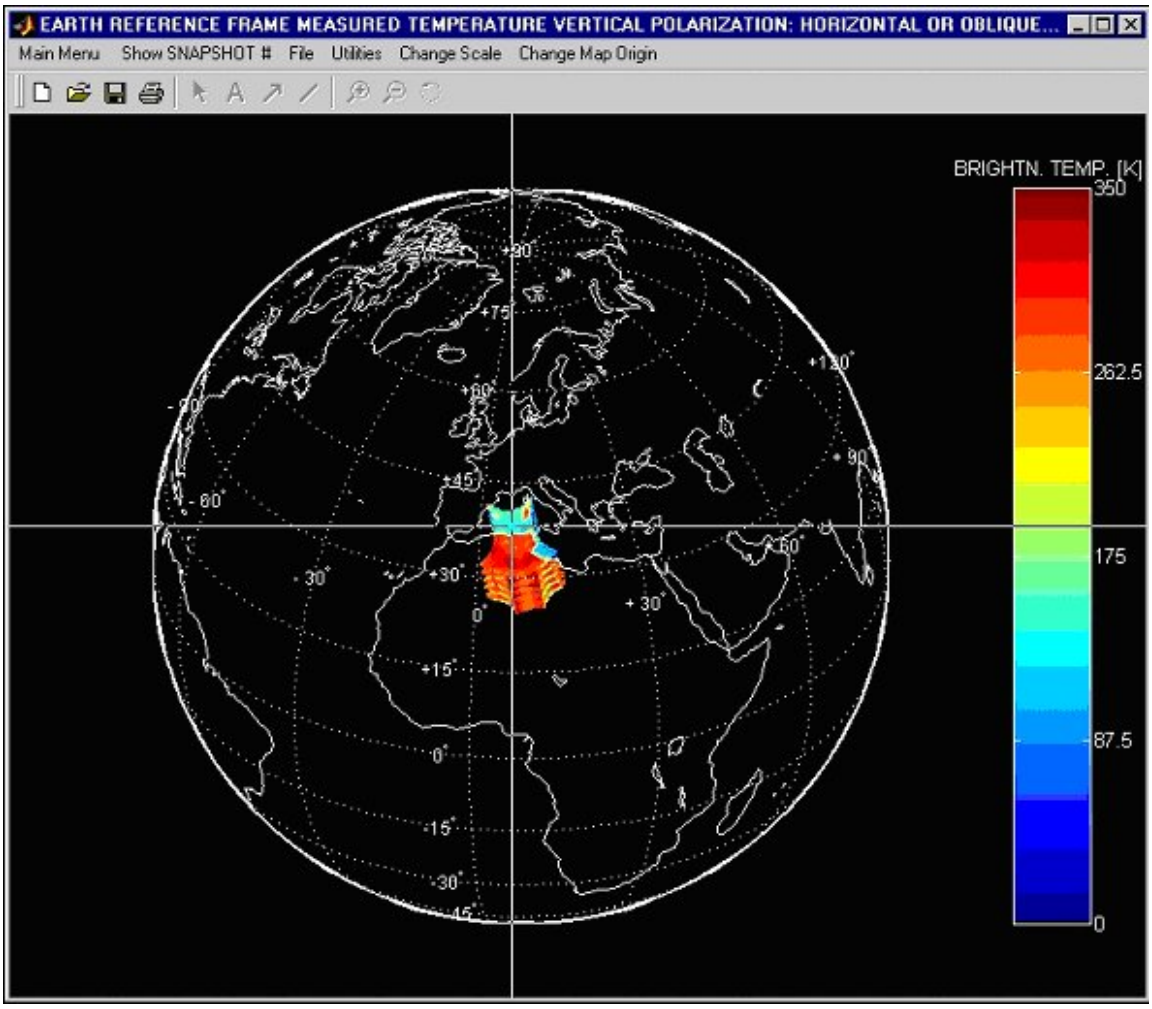


Figure VII.42. The Cursor Position tool allow the user to know the value associated with the selected pixel

Then, the figure's title displays the geographic latitude and longitude in sexagesimal degrees for the selected pixel, and also the brightness temperature value for the current type of data. The following image shows the results.

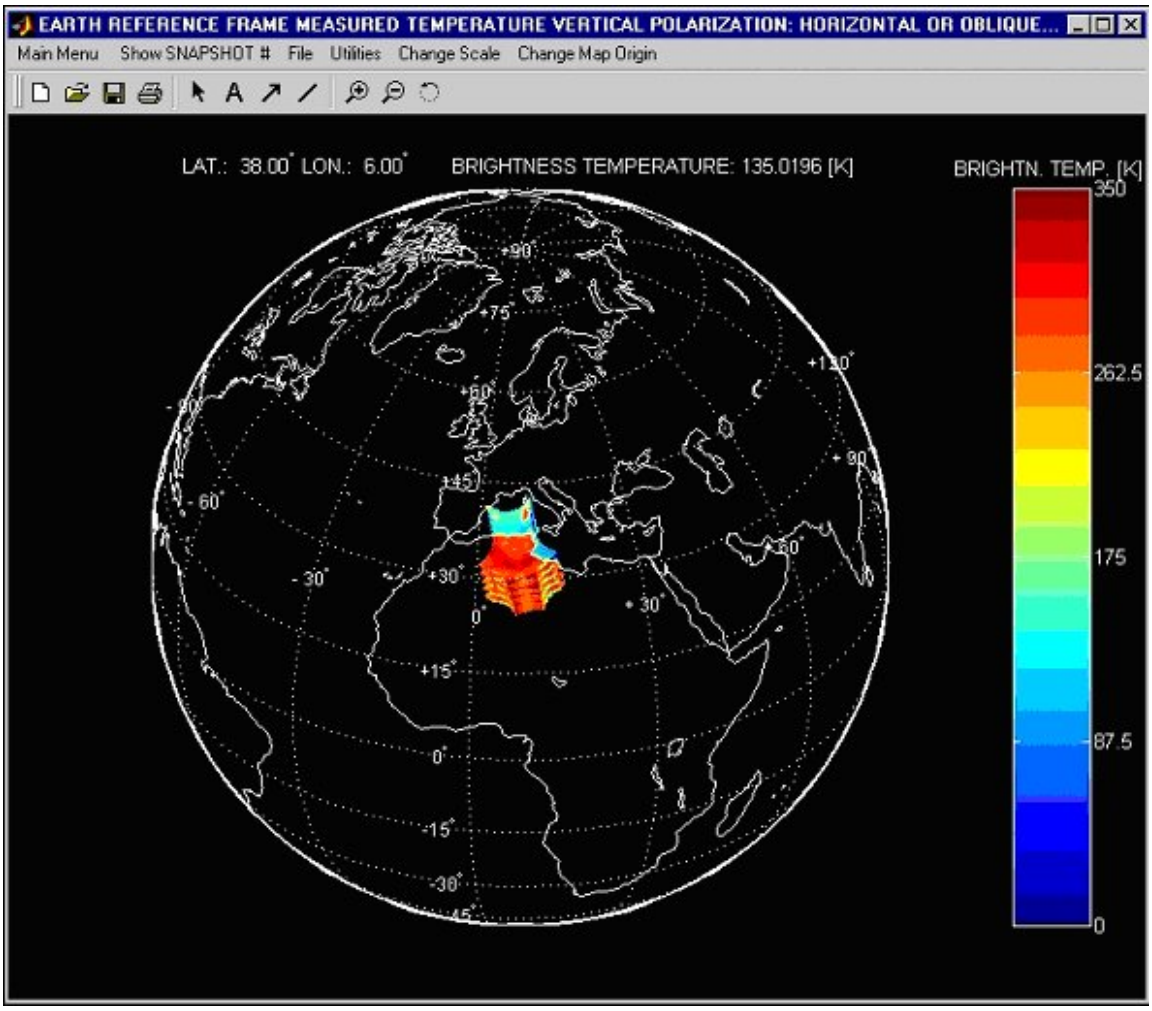


Figure VII.43. The geographic co-ordinates and also the value of the projected variable are displayed in the plot's title

VII.3.2.1.6. Axes On/Off

After clicking the Utilities menu label, the ***Axes*** menu allows the user to visualize the projection in a more comfortable way through three possible alternatives. To turn on or off the visualization of the axes, the meridian labels or the parallel labels. The ***Axes On/Off*** label appears when the user presses the *Axes* menu label.

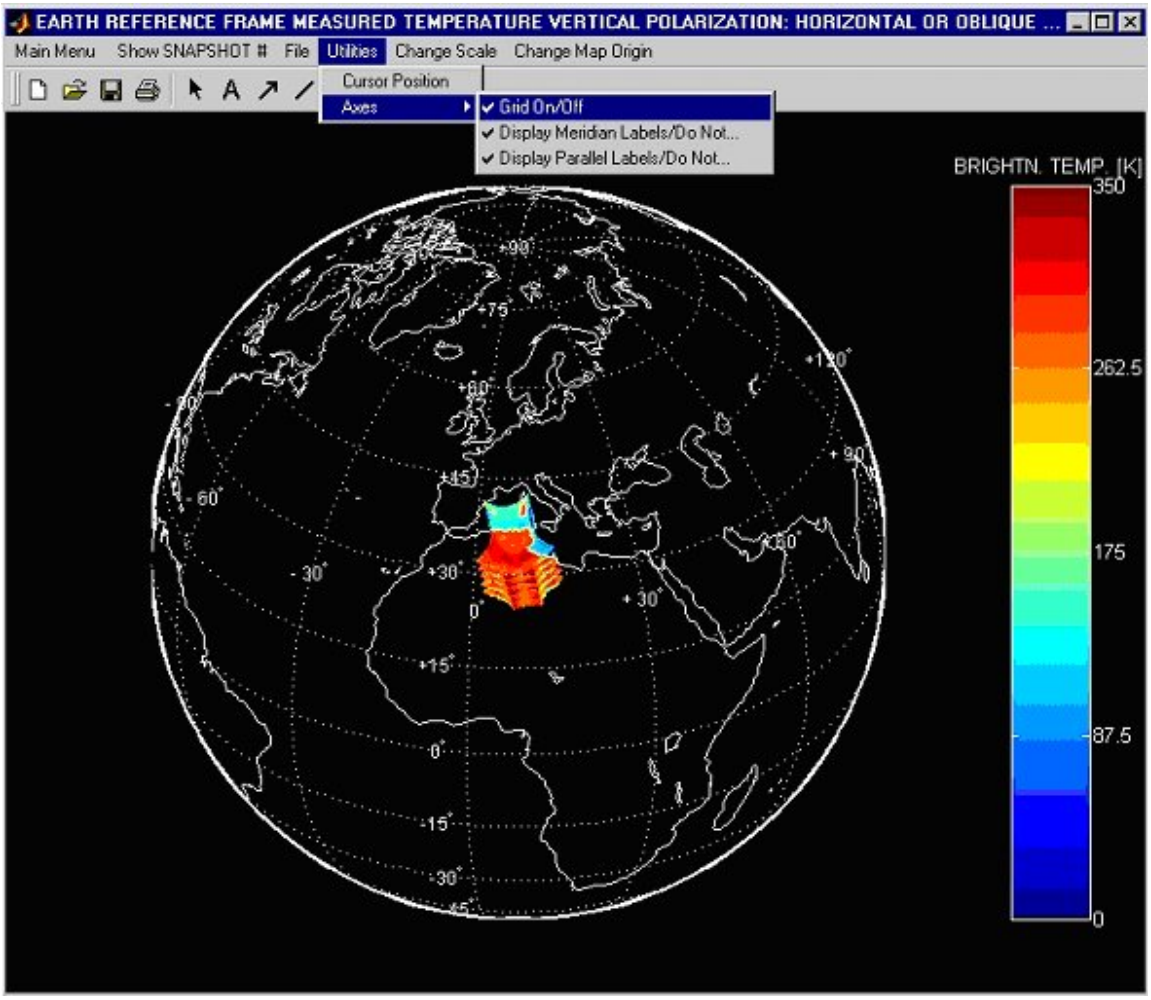


Figure VII.44. The geographic co-ordinates and also the value of the projected variable are displayed in the plot's title

Clicking the mouse on that label, the present meridians and parallels, their labels and the projection frame are displayed or not. The image in Figure VII.45 shows the resulting figure when this label is turned to off.

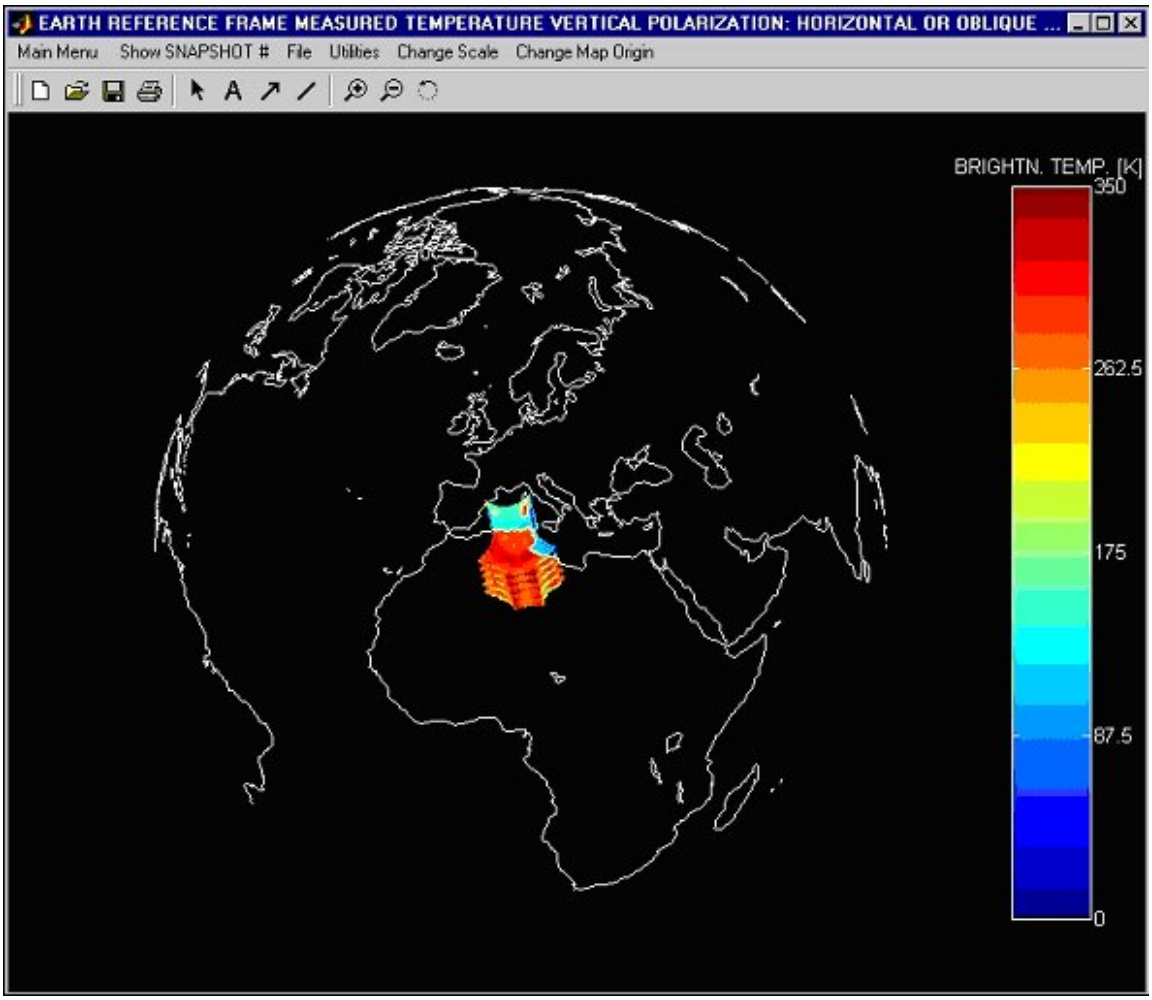


Figure VII.45. The meridians, the parallels and the projection axes can be disabled using the Axes menu label

The ***Display Meridian Labels*** and ***Display Parallel labels*** allow the user to visualize or not the meridian and parallel labels displayed by default. This can be useful for some projections where the accumulation of labels and axes in some zones of the image make the visualization uncomfortable.

VII.3.2.1.7. Change Scale

The scale of the cartographic projected maps can be modified in order to fit the data in a range that is more comfortable for the user. The user may press the ***Change Scale*** label in the figure's menu bar. Tree options are then available:

- ***Auto Scale***: the maximum and minimum values of the images in the figure become the limits of the color scale;
- ***Fixed Scale***: the limits of the scale are fixed, particularly between 0 and 350 K for the absolute value images, and between -10 and 10 K for the error images. Also the limits in the Fixed Scale mode for other variables that can be plotted in the Cartographic Maps domain are defined.
- ***User-selected Scale***: the window presented in Figure VII.46 allows the user to supply the required values for the maximum and minimum of the new scale, in the edit boxes.

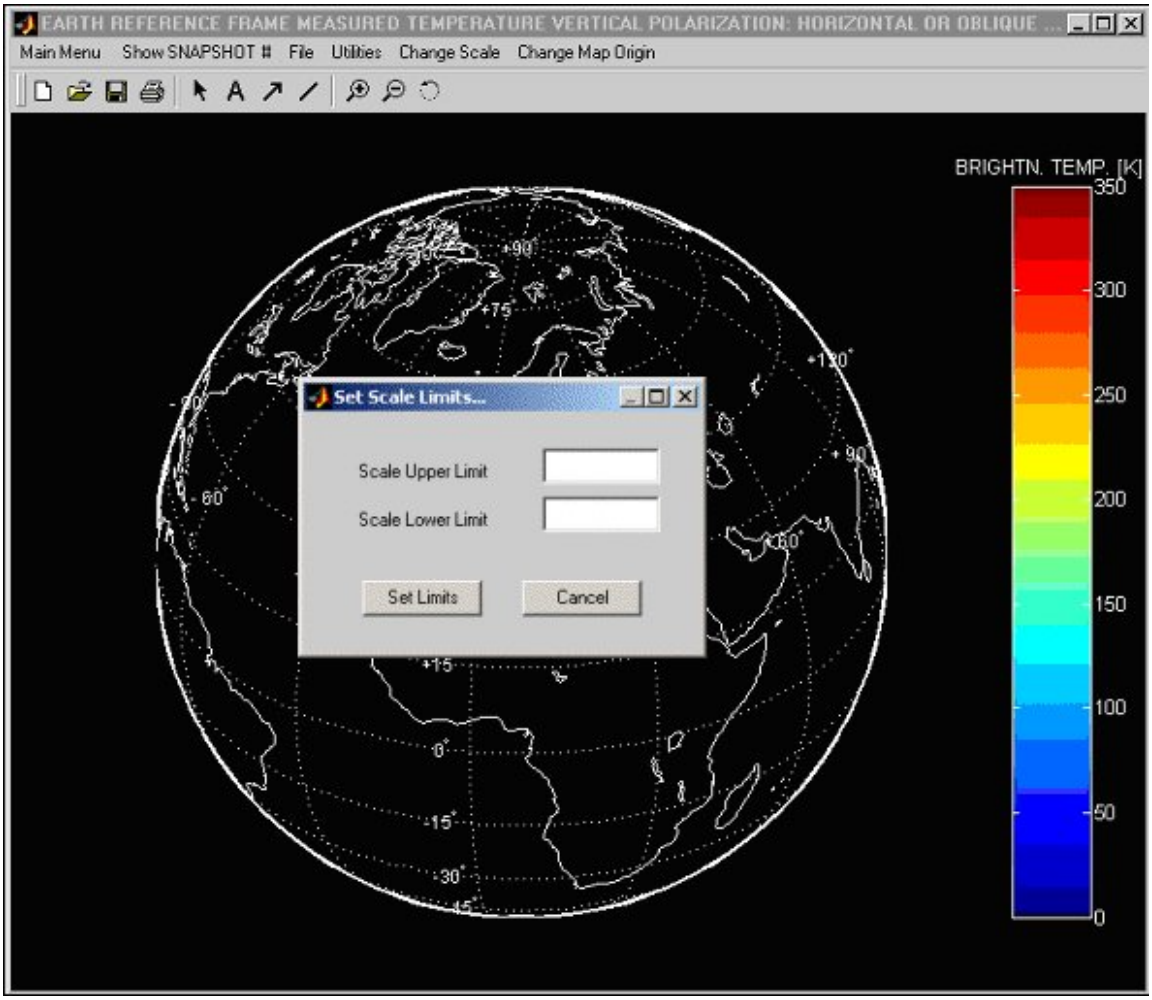


Figure VII.46. The scale of the projected map can be changed using the Change Scale tool

Figure VII.47 shows the resulting figure of limiting the range of the projected image between 100 and 300 Kelvin. Notice the color of the images has changed accordingly.

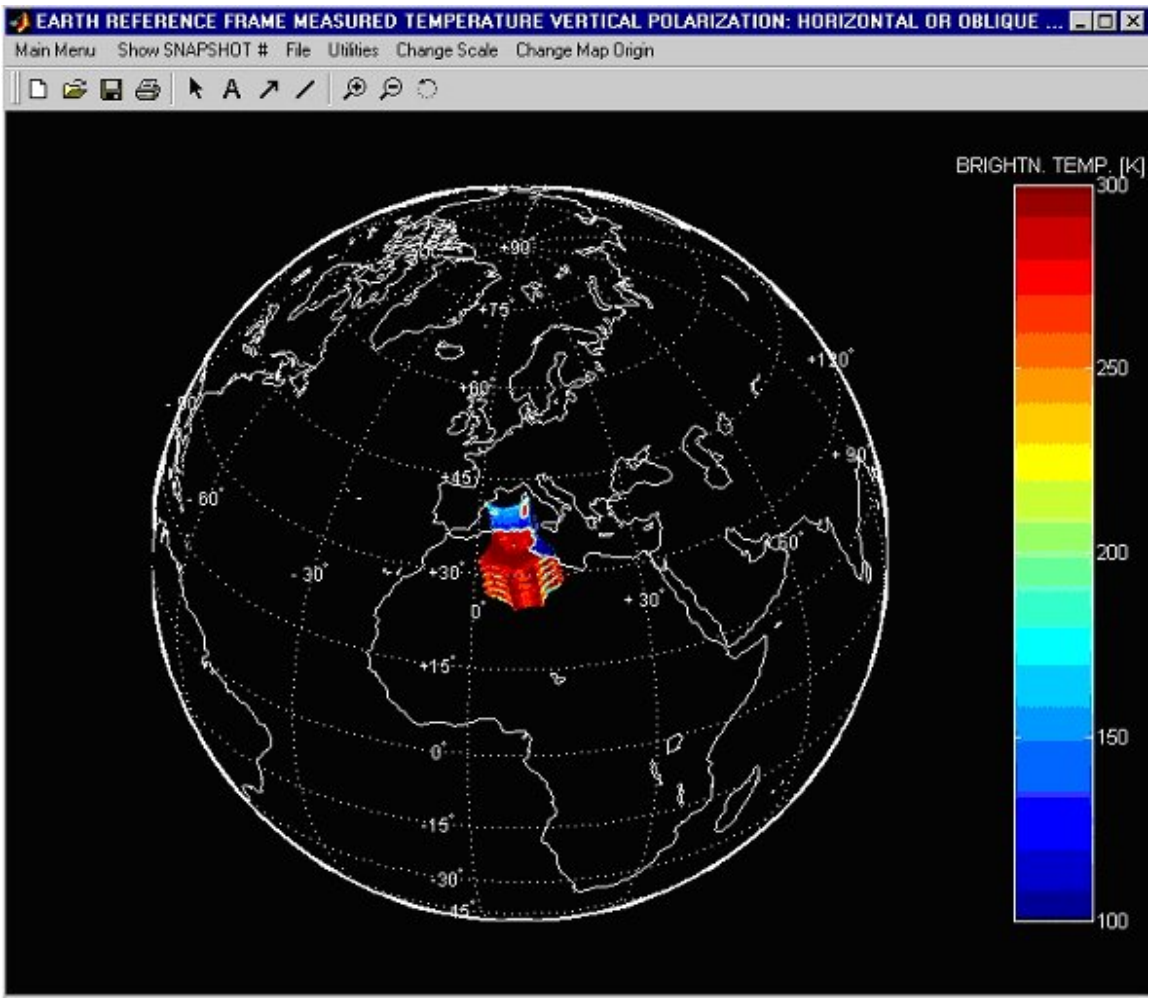


Figure VII.47. The scale is now limited between 100 and 300 Kelvin

The user should be careful to select the new scale limits so as to include the pixels of interest. Placing too narrow limits may cause some pixels to be left outside the scale. Pixels that have brightness temperature beyond the limits will have the same color as the pixels with the limit value. By default the lower limit has color black, and hence all pixels below the lower limit will not be distinguishable from the background.

VII.3.2.1.8. Change Map Origin

The center of the projection is by default the geographic center of the projected image. This is good for data concentrated in a small geographic area, but when the simulated scenario covers a large geographic area, it is useful that the observer's point of view can be changed. The ***Change Map Origin*** tool allows it. Pressing this label in the window's menu bar, the small dialog shown in Figure VII.48 is then opened.

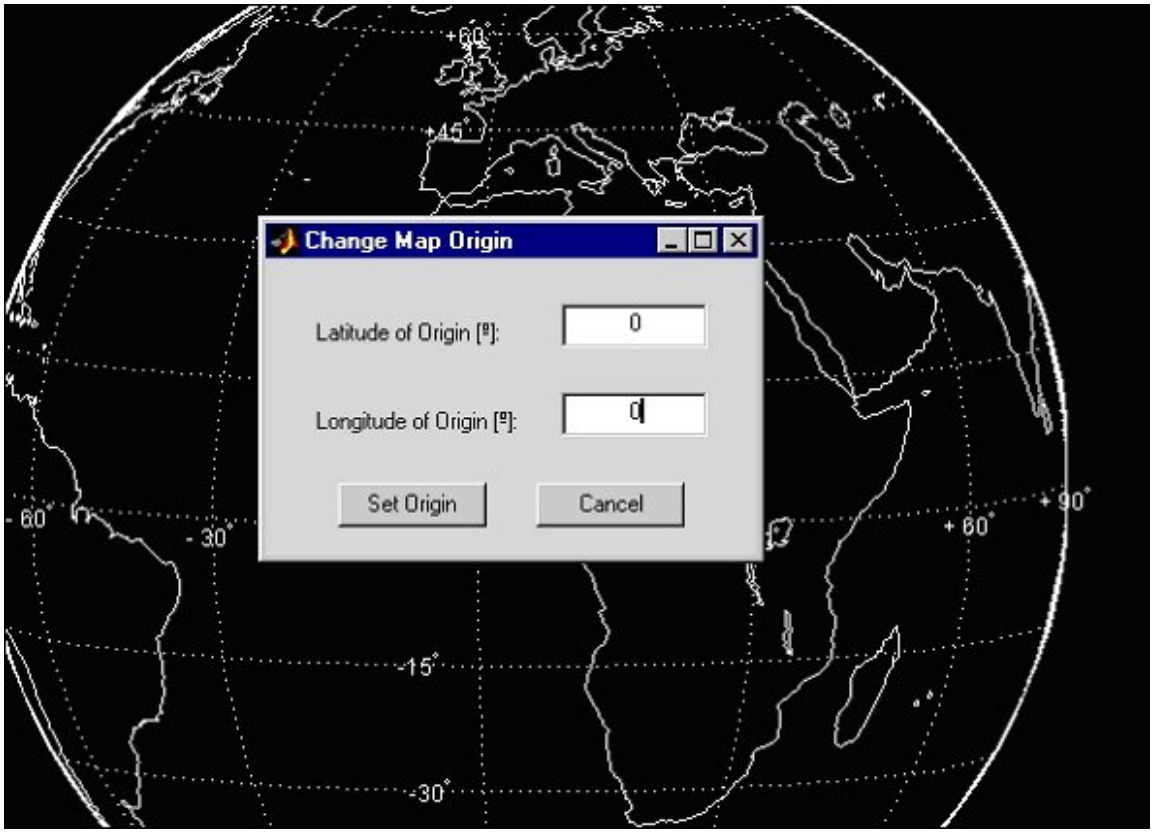


Figure VII.48. The latitude and longitude of the new origin must be supplied by the user

Figure VII.49 shows the resulting figure for an origin of 0º latitude and 0º longitude.

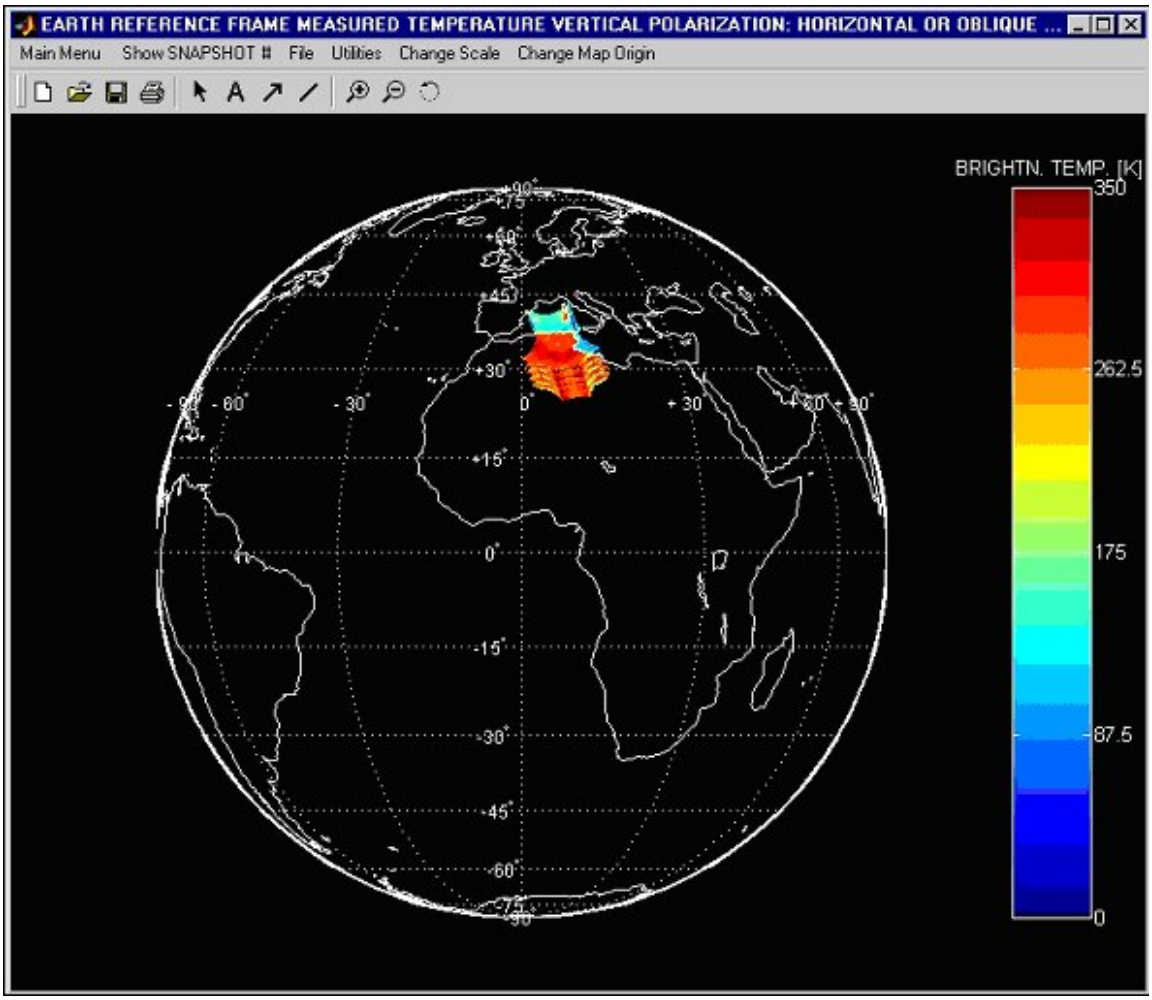


Figure VII.49. The map origin is now in 0º latitude and 0º longitude

VII.3.2.1.9. Zoom

The ***Zoom*** tool allows the user observing the projected map closer. Through the other menu tools the user has the possibility to obtain a pixel's brightness temperature value through the *Cursor Position* label selecting the pixel more precisely. For executing a zoom, the user may click on the magnifying glass shown in the figure's menu bar, then select an area of the image holding the mouse left button while describing a rectangle enclosing it. The next image shows the effect of performing a *Zoom*.

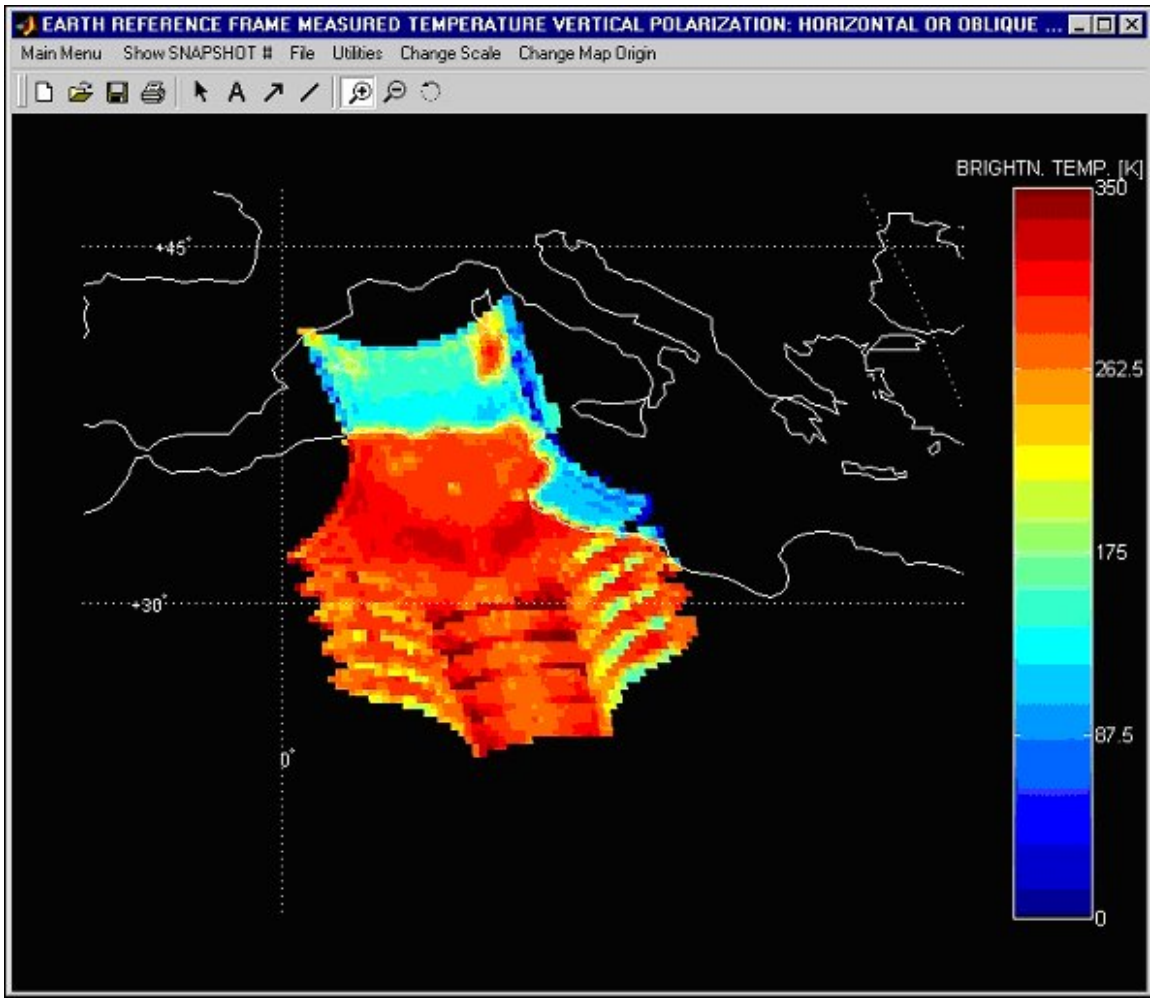


Figure VII.50. The effect of performing a zoom over the projected map

VII.3.2.2. Projections

This section overviews the main characteristics of the projections available to the user through the *Cartographic Representation Method* window presented at the beginning of section VII.3.2.1.1. Representative images are also included to make the choice of a particular projection easier, following the example introduced in the previous section. For more information on this topic the user may consult the AD/DD document (SO-TN-GMV-PLM-0003), section 6.57 *FOM4: FOM and Statistical Analysis. Brightness Temperature Maps*.

VII.3.2.2.1. Gnomonic Projections

This is a perspective projection from the center of the Earth on a plane tangent at the center point. The first three buttons in the *Cartographic Representation Method* window are available to the user for performing the *Gnomonic* projections. The difference between these projections is the origin latitude of the projection axes. The first button opens a figure where the ***Equatorial or Direct Gnomonic Projection*** is performed. The axes' origin is at 90° latitude and the center longitude of the plotted image. Figure VII.51 next image shows the resulting figure.

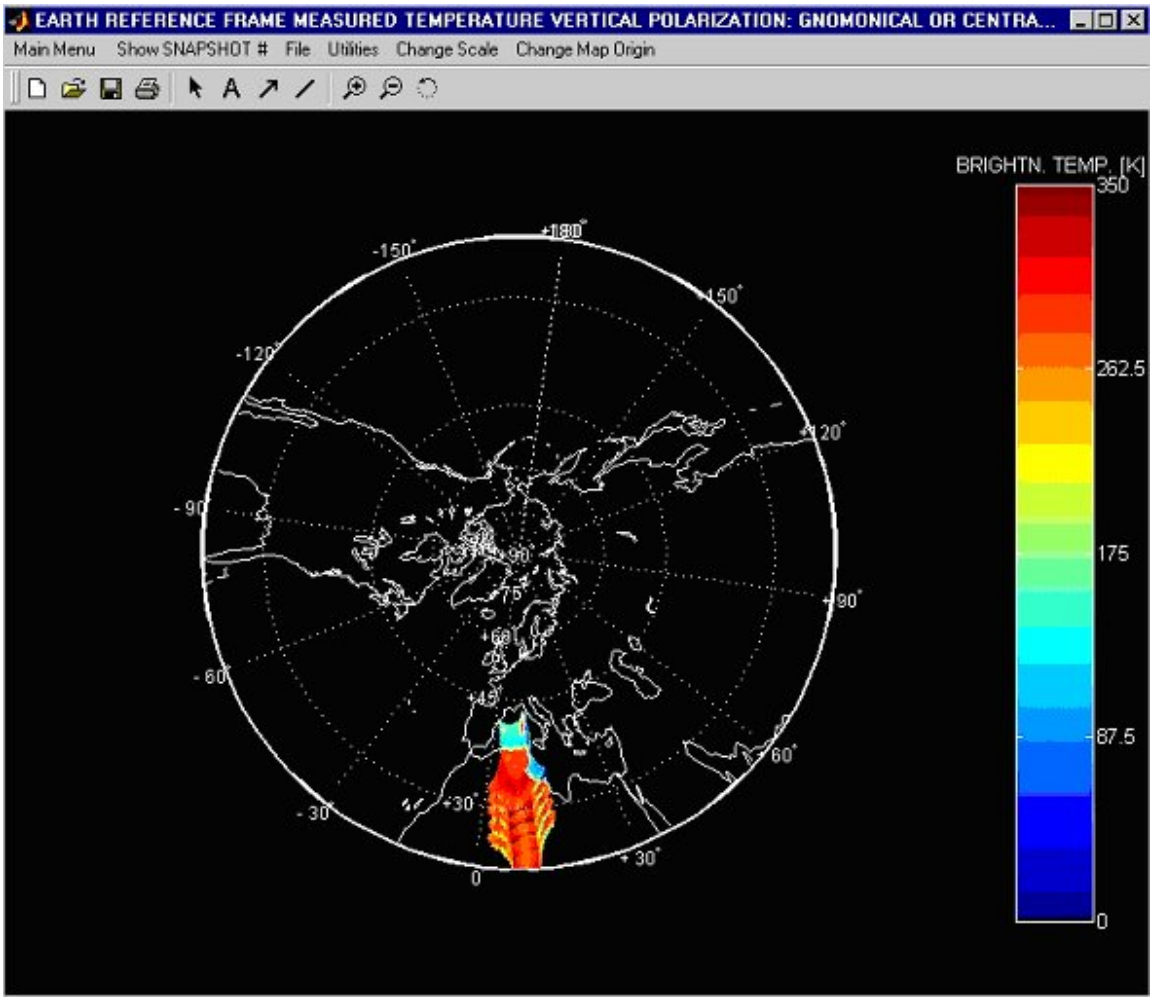


Figure VII.51. Equatorial or Gnomonic projection

The second button opens a figure where the ***Meridian or Transverse Gnomonic Projection*** is performed. The axes' origin is at 0° latitude and the center longitude of the plotted image. The third button opens a figure where the ***Horizontal or Oblique Gnomonic Projection*** is performed. The axes' origin is at the center latitude and longitude of the plotted image.

VII.3.2.2.2. Stereographic Projections

This is a perspective projection on a plane tangent at the center point from its antipodal point. The second group of three buttons in the *Cartographic Representation Method* window is available to the user for performing the *Stereographic* projections. The difference between these projections is the origin latitude of the projection axes.

The first button of this group opens a figure where the ***Polar Stereographic Projection*** is performed. The axes' origin is at 90° latitude and the center longitude of the plotted image. Figure VII.52 shows the resulting projected map.

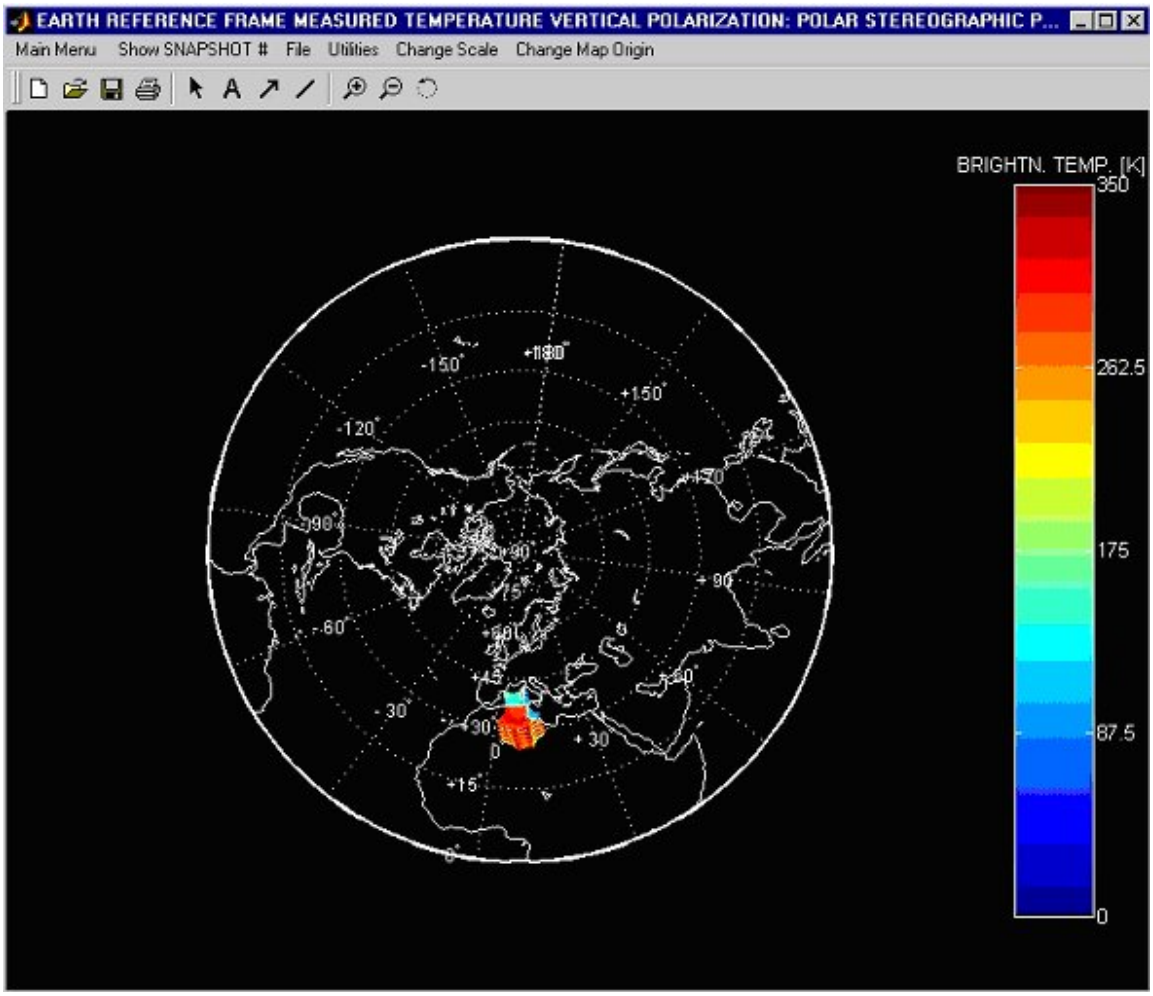


Figure VII.52. Polar Stereographic projection

The second button of this group opens a figure where the ***Meridian or Transverse Stereographic Projection*** is performed. The axes' origin is at the 0° latitude and the center longitude of the plotted image. The third button opens a figure where the ***Horizontal or Oblique Stereographic Projection*** is performed. The axes' origin is at the center latitude and longitude of the plotted image.

VII.3.2.2.3. Orthographic Projections

This is a perspective projection on a plane tangent at the center point from an infinite distance. The third group of three buttons in the *Cartographic Representation*

Method window is available to the user for performing the *Orthographic* projections. The difference between these projections is the origin latitude of the projection axes.

The first button of this group opens a figure where the ***Equatorial or Direct Orthographic Projection*** is performed. The axes' origin is at 90° latitude and the center longitude of the plotted image. Figure VII.53 shows the resulting projected map.

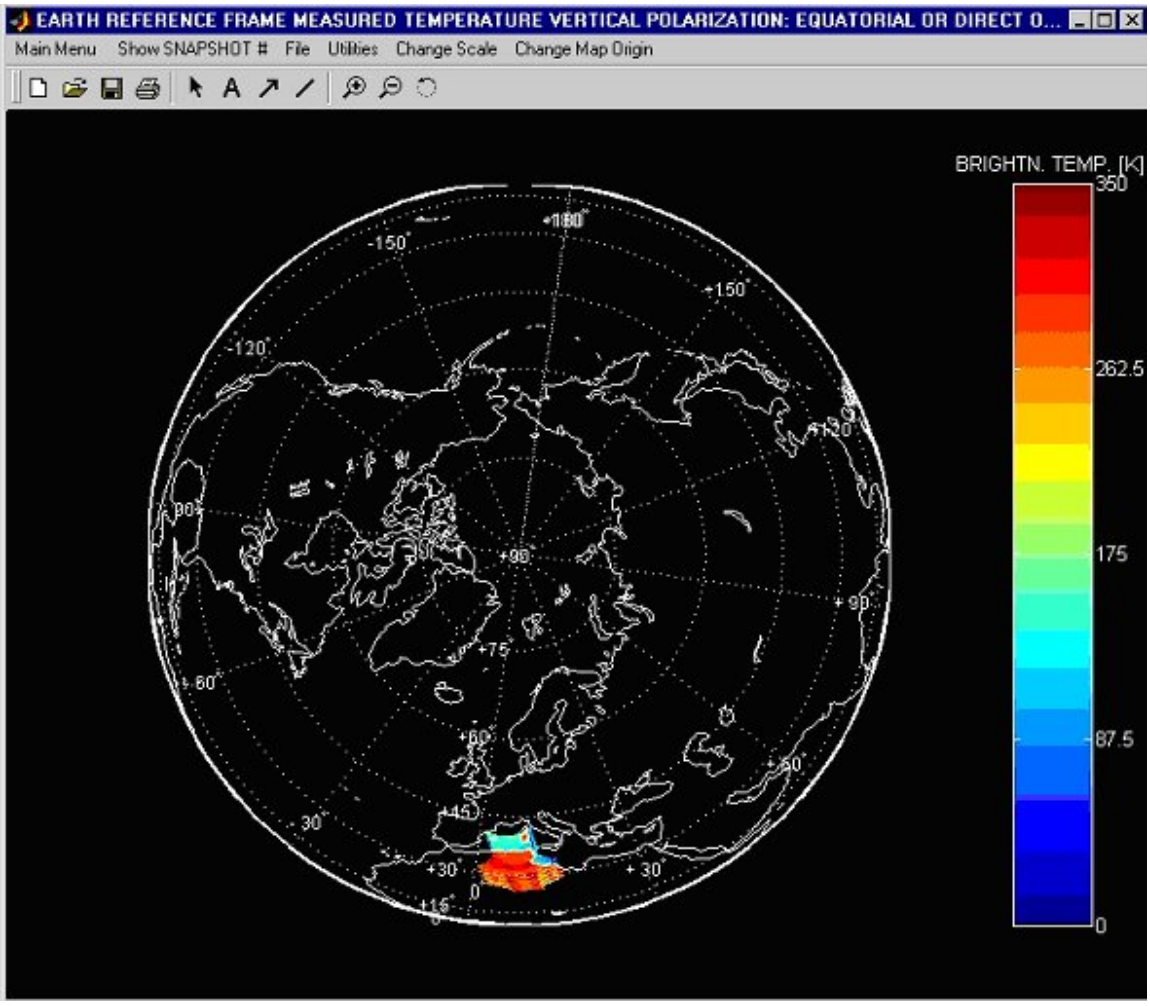


Figure VII.53. Equatorial or Direct Orthographic projection

The second button of this group opens a figure where the *Meridian* ***or Transverse Orthographic Projection*** is performed. The axes' origin is at the 0° latitude and the center longitude of the plotted image. The third button of this group opens a figure where the

Horizontal or Oblique Orthographic Projection is performed. The axes' origin is at the center latitude and longitude of the plotted image.

VII.3.2.2.4. Cylindrical Developments

The fourth group of buttons in the *Cartographic Representation Method* window shows data using cylindrical developments. There are five possible cylindrical developments.

The first button of this group opens a figure where the ***Lambert's Cylindrical Development*** is performed. This is an orthographic projection onto a cylinder tangent at the Equator. Figure VII.54 shows the projected map.

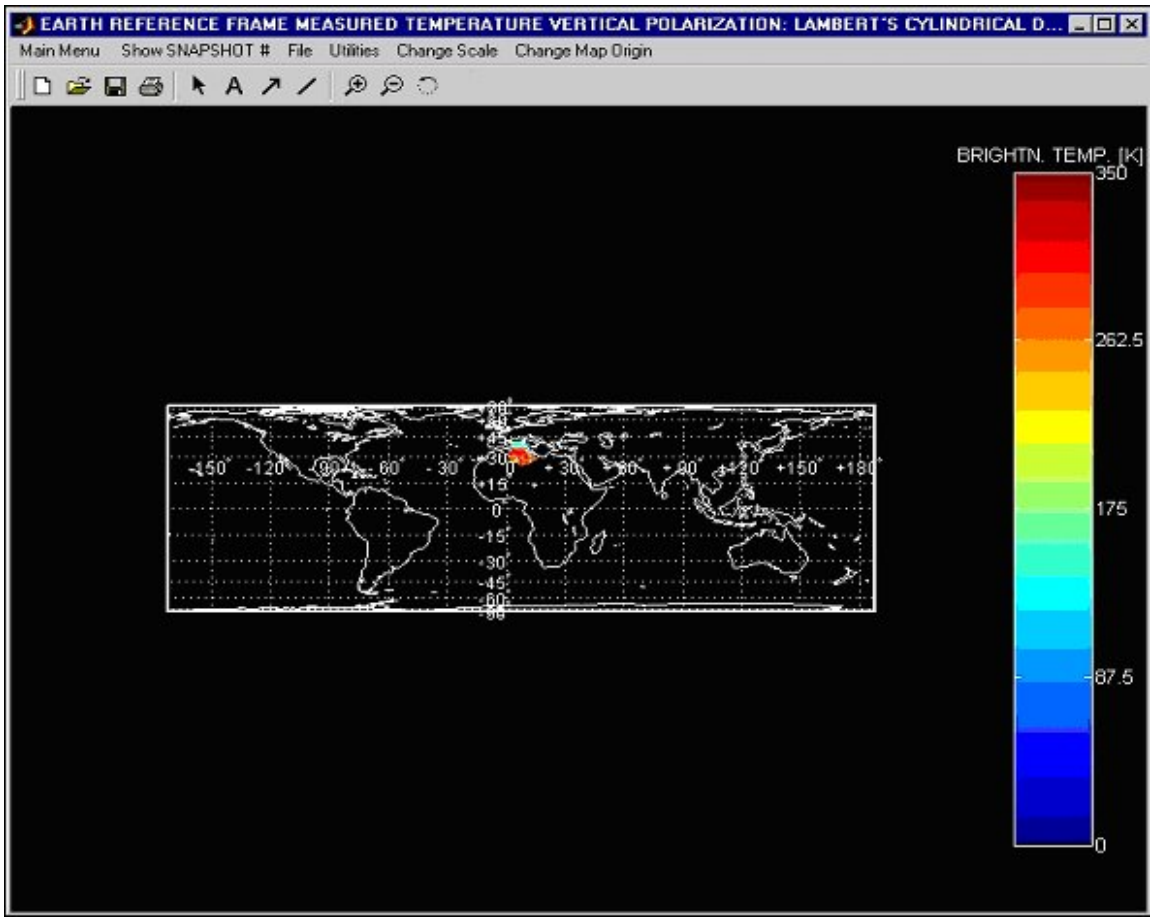


Figure VII.54. Lambert's Cylindrical Development

The second button of this group opens a figure where the ***Cylindrical Development with Automecoic Meridians*** is performed. This is a projection onto a cylinder secant at the standard parallels. Figure VII.55 shows the projected map.

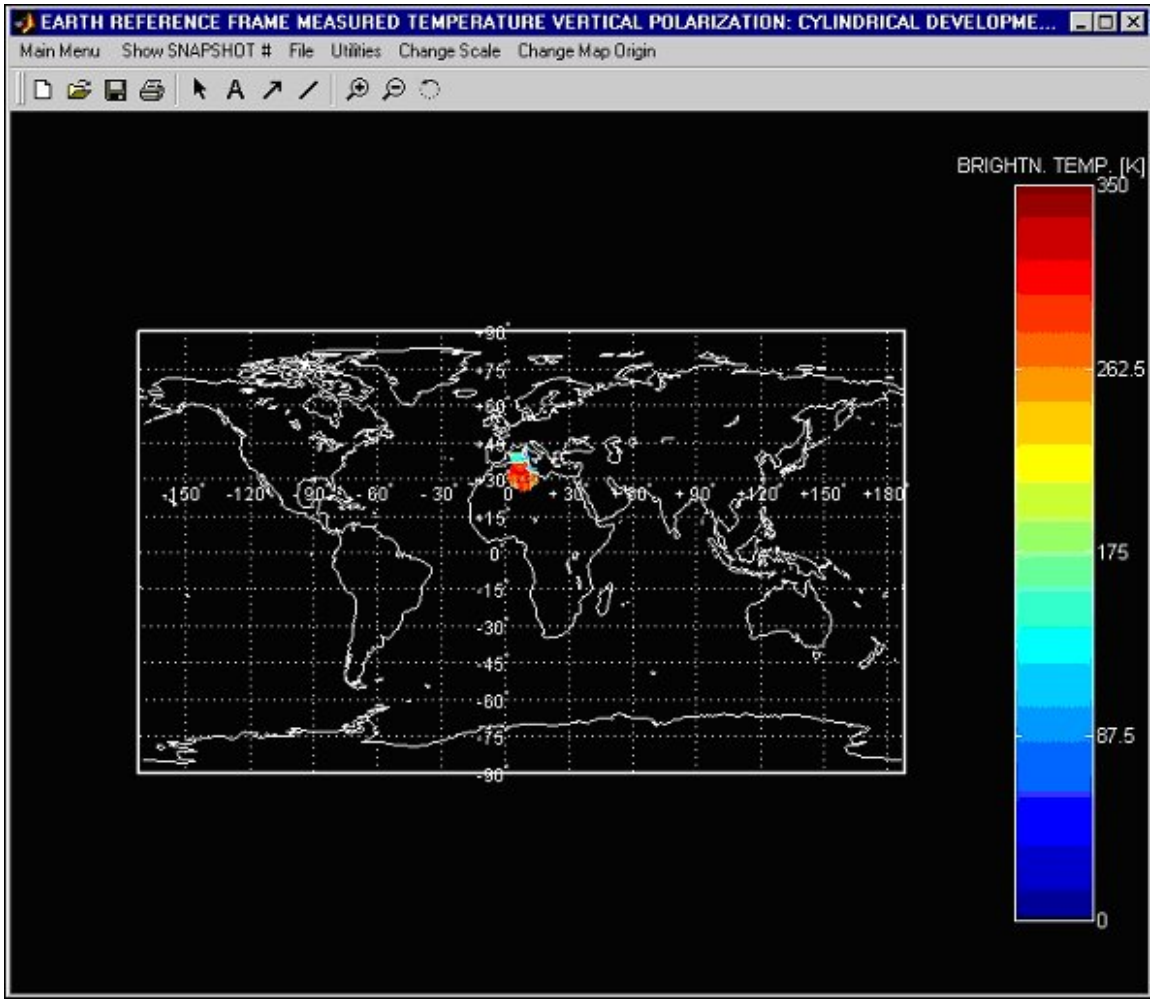


Figure VII.55. Cylindrical Development with Automecoic Meridians

The third button of this group opens a figure where ***Consistent Cylindrical Development (Mercator's Chart)*** is performed. It is a conformal projection. Figure VII.56 shows the projected map.

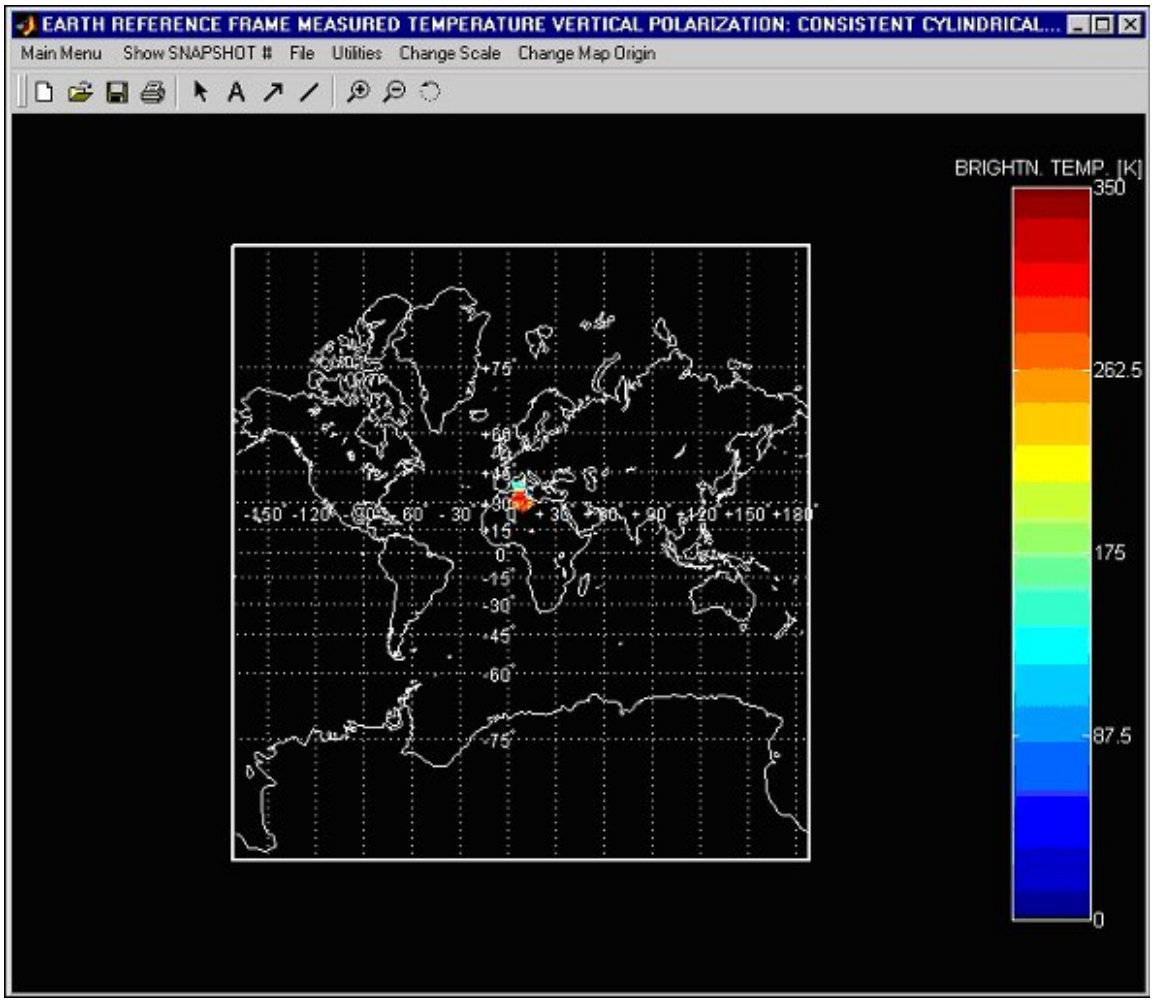


Figure VII.56. Consistent Cylindrical Development (Mercator's Chart)

The fourth button of this group opens a figure where ***Mercator's Cylindrical Development*** is shown. It is almost equal to the above development, but its geoid is defined by the Earth's semi-major axis and eccentricity. Figure VII.57 shows the projected map.

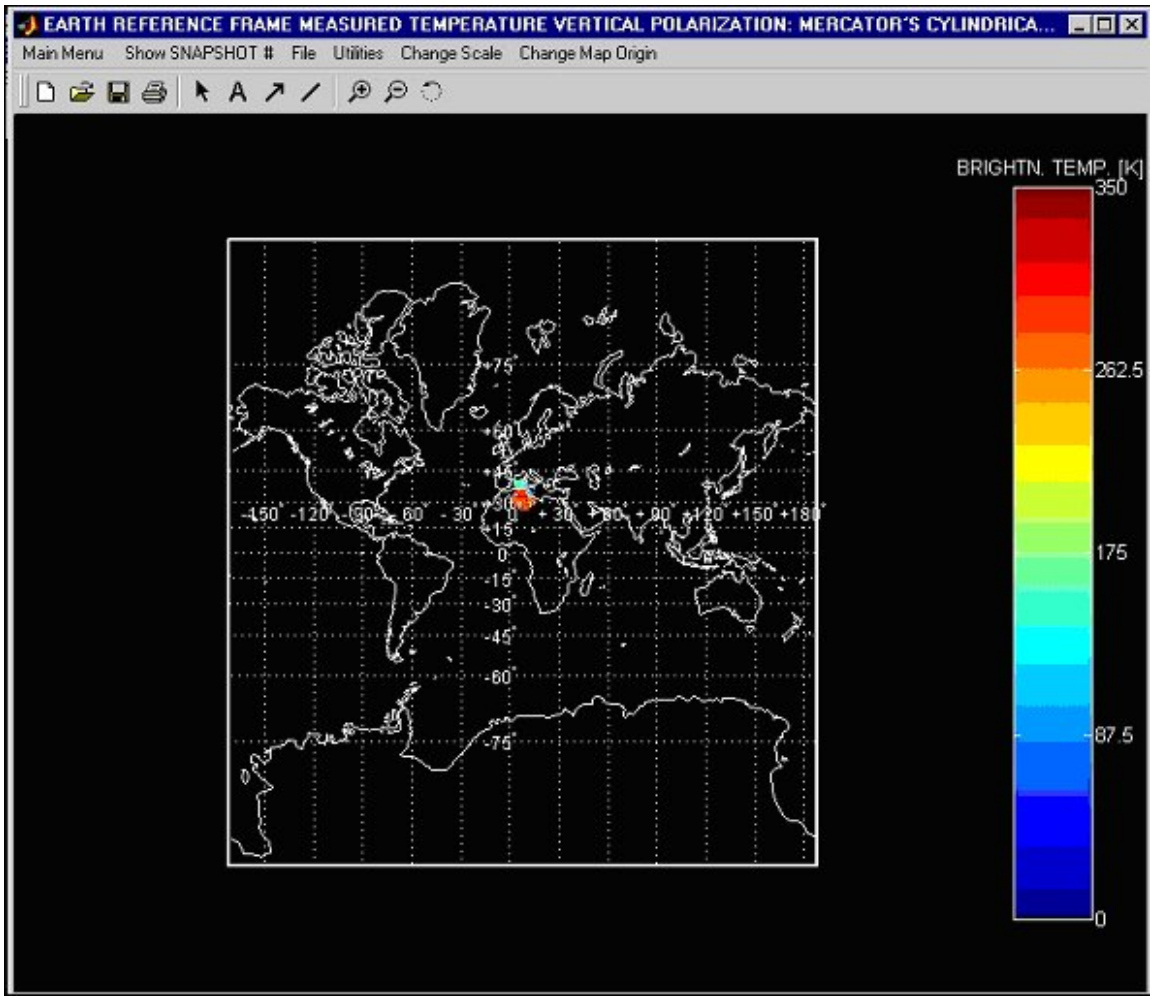


Figure VII.57. Mercator's Cylindrical Development

The fifth button of this group opens a figure where the ***Gauss' Transverse Cylindrical Development*** is performed. It is a projection onto a cylinder tangent at the central meridian. Figure VII.58 shows the projected map.

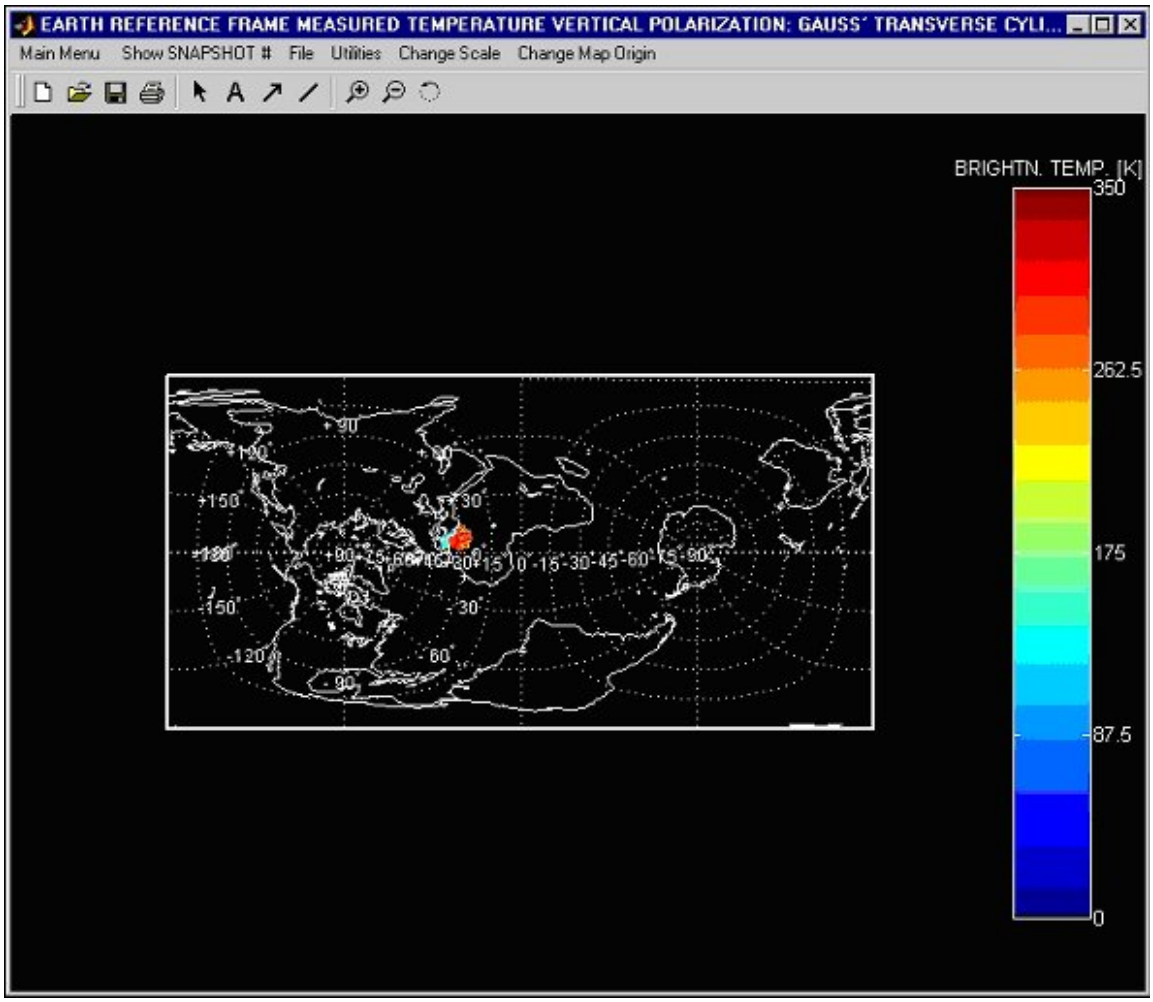


Figure VII.58. Gauss' Transverse Cylindrical Development

VII.3.2.2.5. Universal Transverse Mercator Projection

This is a projection designed to minimize distortion over a defined set of small areas. It preserves angles, but it does not preserve areas or distances. It is only defined for latitudes below 84° North and above 80° South.

Pushing the ***UTM Projection*** button, two figures are opened. The first one displays an *UTM* map whose geographic limits are fitted to the displayed data. The *UTM* zones are unfolded along their central meridians, as if they were the unfolded skins of the orange's segments. The user may note that the grid lines define the centers of the *UTM* zones, not

the limiting curves of the *UTM* zones. This figure is just opened to give the user a global view of the scene, but as it is not comfortable enough, the possibility of selecting singles *UTM* zones is given. Therefore, the menu labels to handle the image are available only in the single-zone figures. Figure VII.59 shows the *UTM* global map.

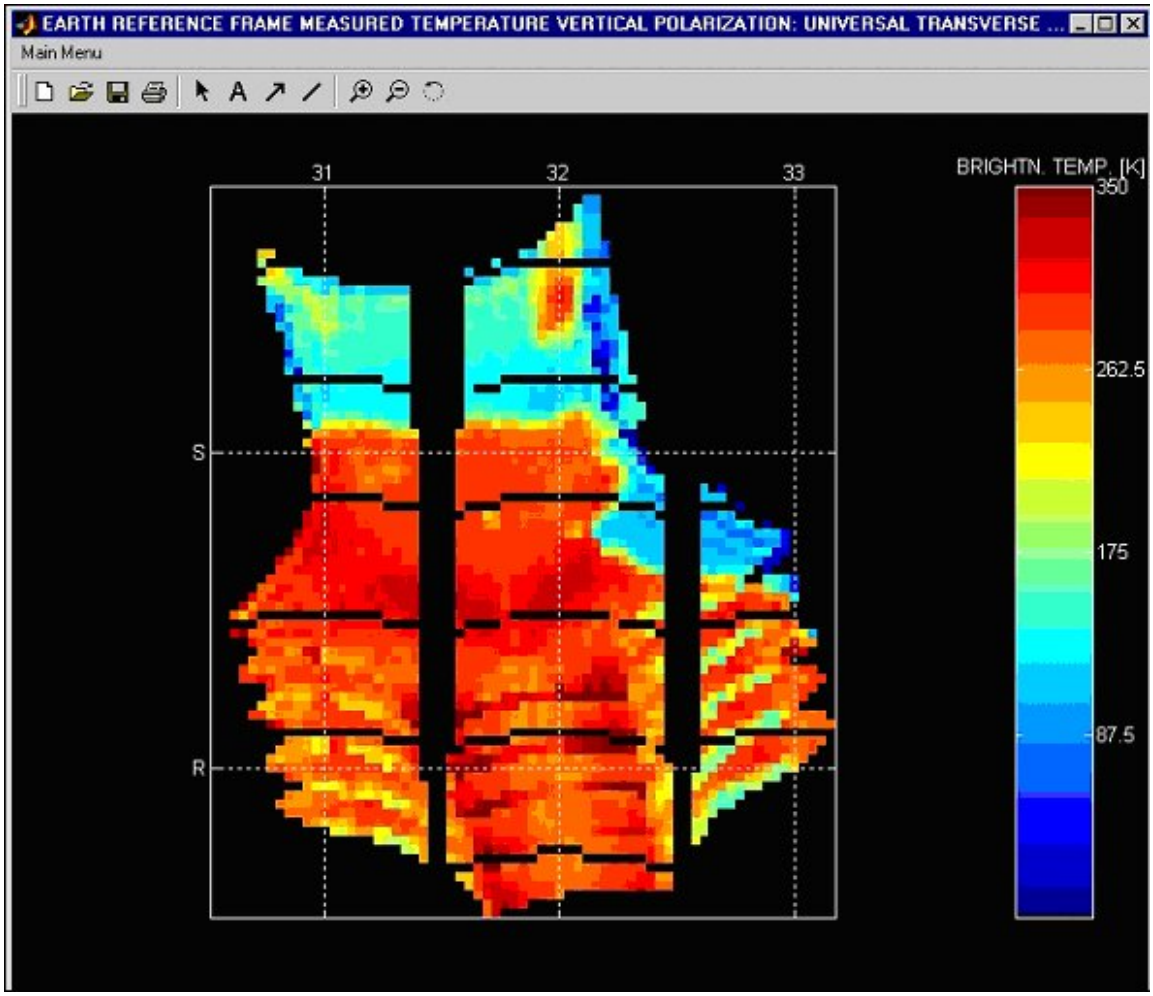


Figure VII.59. Global map of the U.T.M. Projection

The second opened figure is the ***Selection*** figure. This figure gives the user the possibility to enter the identifying string for a single *UTM* zone. Figure VII.60 shows its appearance.

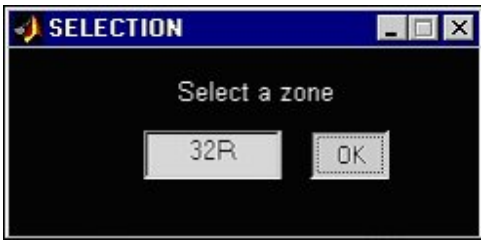


Figure VII.60. Selection figure to choose one U.T.M. zone to plot

Once the string for the selected zone is entered, the user must press the ***OK*** button to open a third figure where the requested *UTM* zone is projected. As it can be observed in the following image, this single-zone figure allows the user to perform all the actions available for the other projections (*Cursor Position*, *Zoom*, etc…).

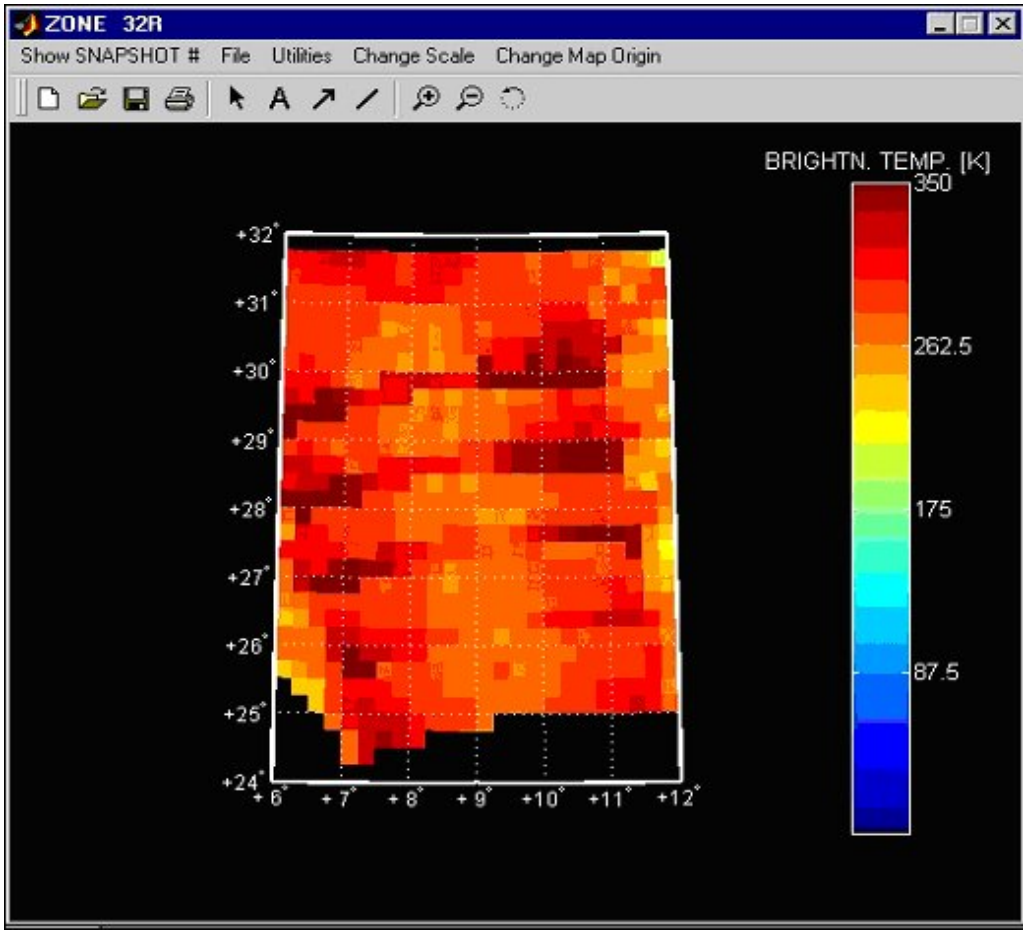


Figure VII.61. Selected U.T.M. zone -32R in this example.

VII.3.2.2.6. Conical Developments

The last group of buttons in the *Cartographic Representation Method* window allows the user to perform Lambert's conical developments. The Lambert's conical developments preserve angles everywhere but the poles. They do not preserve areas or distances. Figure VII.62 shows the ***Lambert's Consistent Conical Development***.

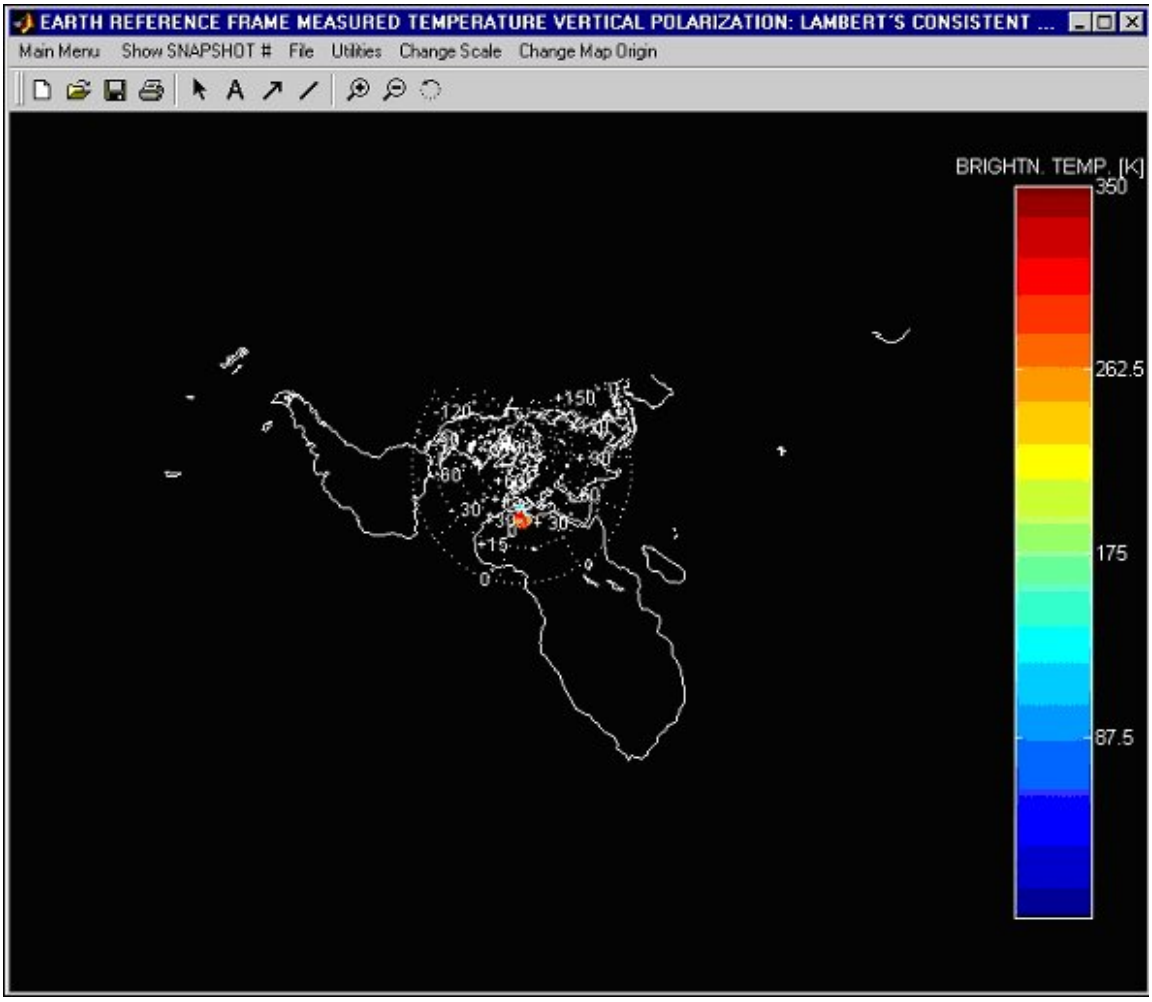


Figure VII.62. Lambert's Consistent Conical Development.

Figure VII.63 displays an example of the ***Strictly Lambert's Consistent Conical Development***, which in fact applies the same projection as for the ***Lambert's Consistent Conical Development*** except for the geoid used to perform the projection.

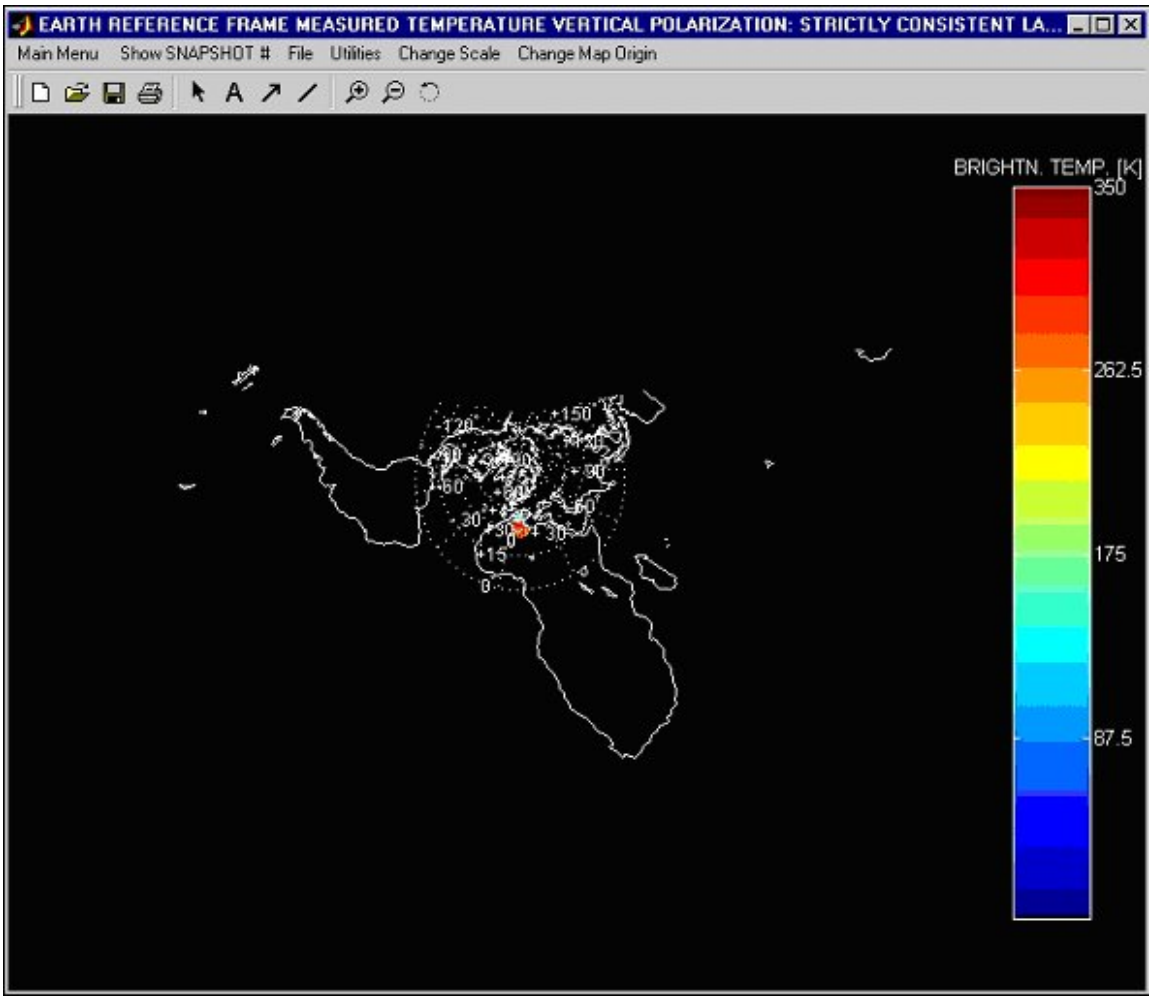


Figure VII.63. Strictly Lambert's Consistent Conical Development.

VII.3.2.3. Types of Data

The next subsections overview the types of data that are available through the *Cartographic Representation Method*. All the attached examples are displayed using the *Horizontal or Oblique Orthographic Projection*.

For further information on this topic, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.57 *FOM4: FOM and Statistical Analysis. Brightness Temperature Maps*).

VII.3.2.3.1. Initial types of data

As it has been said at the beginning of the section VII.3, the data that initially can be plotted consists of brightness temperature data, either obtained from the brightness temperature Generator (***original***), from the computation of the image reconstruction algorithm selected by the user (***retrieved*** *or* ***measured***), or from the computation of an ideal noise-free system using FFTs to retrieve data (***ideal***). Error images can be computed as the differences between these types of data. And all the mentioned data can be plotted either in the antenna's reference frame or in the Earth's reference frame. Figure VII.64 shows the projection of a retrieved or measured temperature image at the Earth's reference frame in vertical polarization.

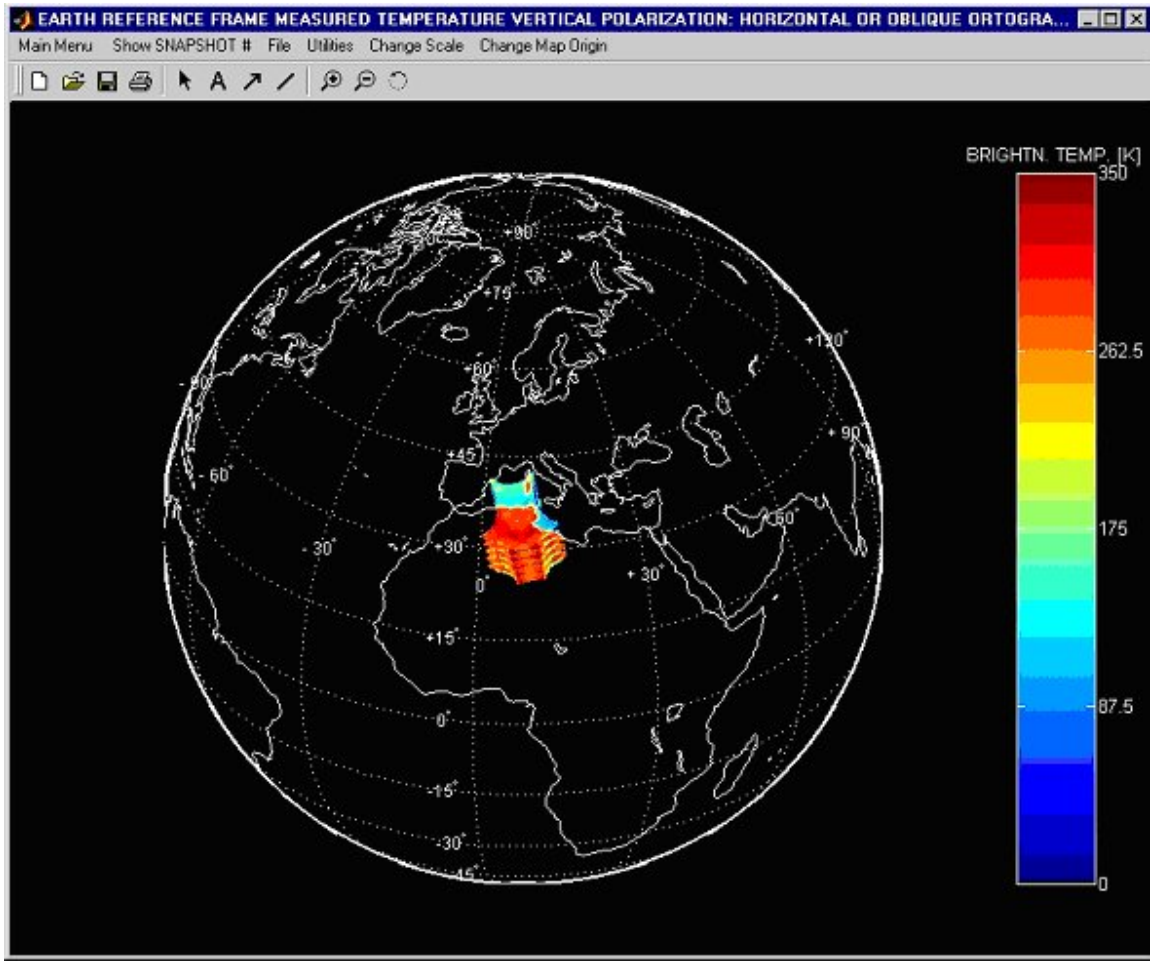


Figure VII.64. Retrieved brightness temperature map on Earth's reference frame, vertical polarization

The chosen type of data is mentioned in the figure's title, as it has been said in previous sections.

VII.3.2.3.2. Projecting other types of data

However, for the SEPS v4.0 some tools have been added to allow the user change the projected data when the projection has been already performed. Figure VII.65 shows the menu that appears when the user clicks with the ***right*** mouse button on the projected image. If the ***Change Parameter*** option is selected, a list of available data to be projected appears.

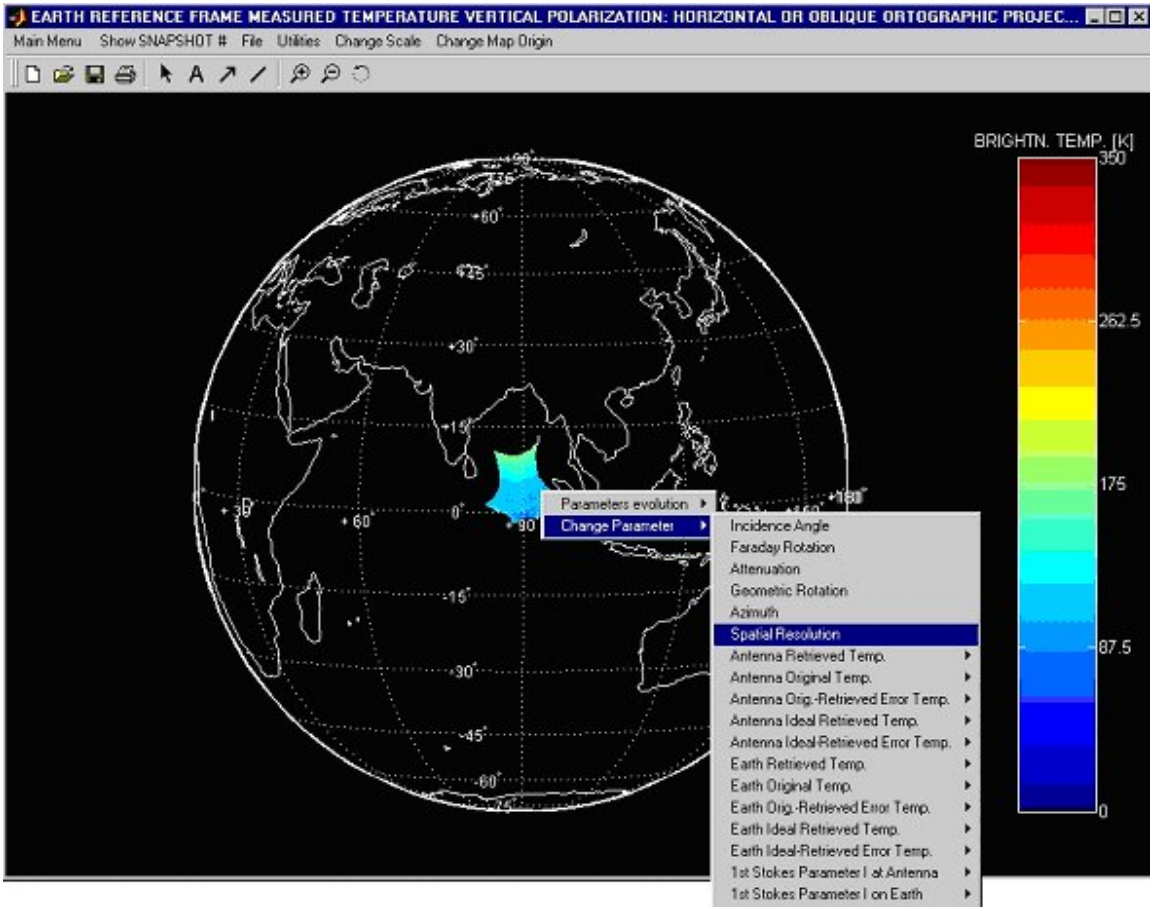


Figure VII.65. Data menu to change the type of data projected on the map

This data consists not only of brightness temperature but also of others parameters, such as the incidence angle, the Faraday rotation, or the spatial resolution. Figure VII.66 shows the result of selecting the Faraday rotation images to be projected on the current map axis. Figure VII.67 shows the equivalent done for spatial resolution data.

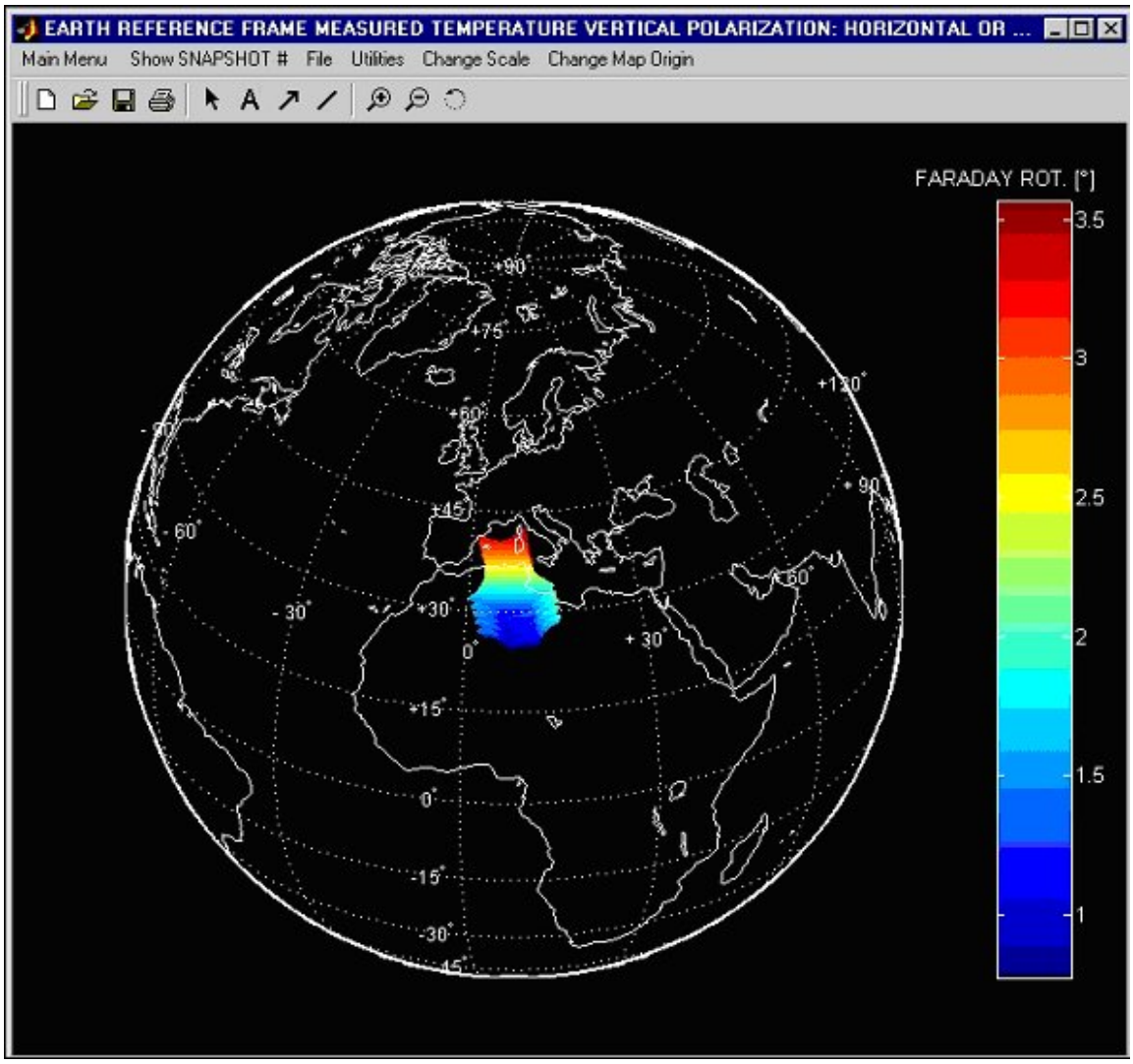


Figure VII.66. Faraday rotation data projected on the current map axis

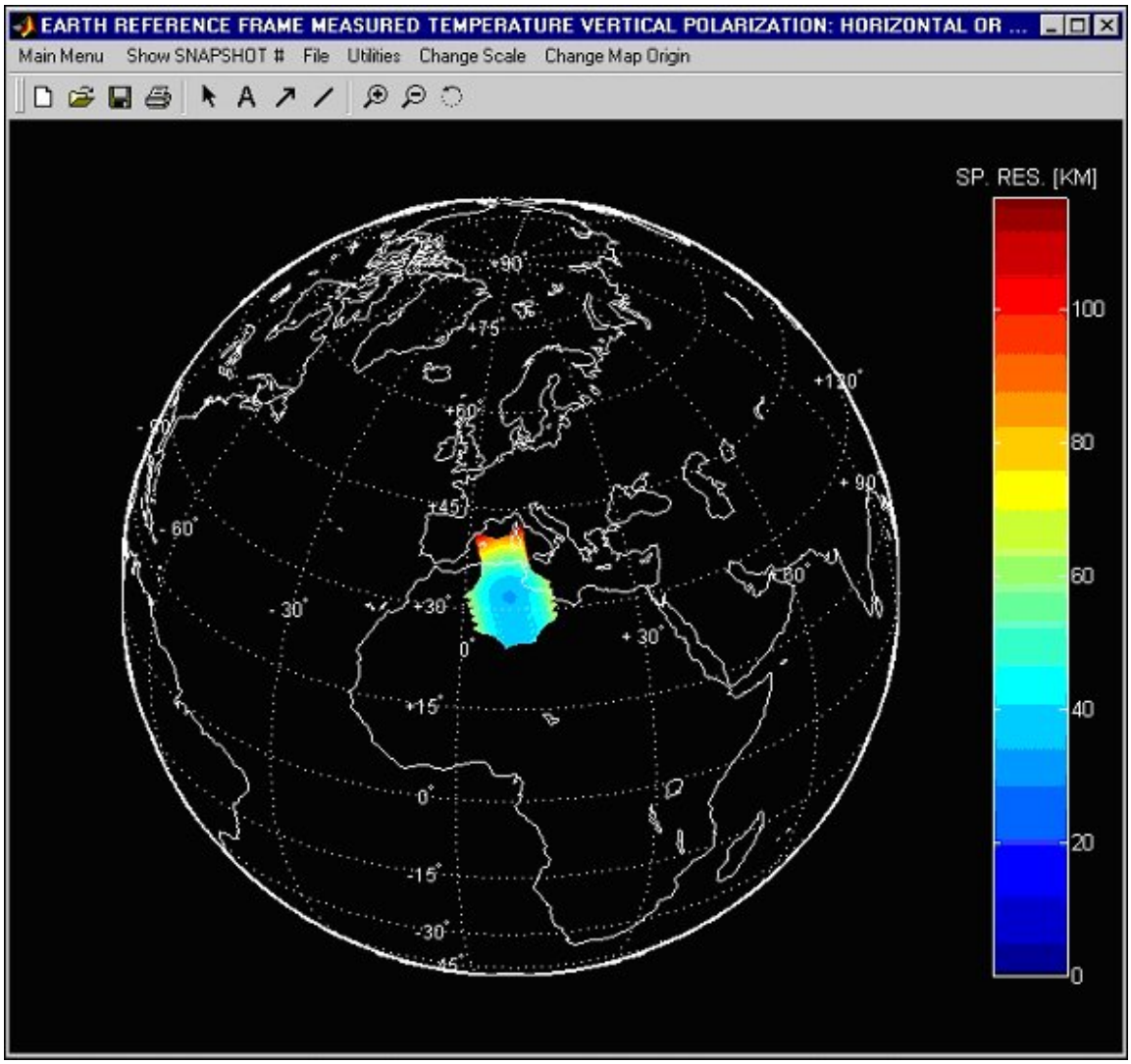


Figure VII.64. Spatial resolution data projected on the current map axis

The user may notice that the type of data being currently plotted is displayed in the scale bar's top text.

VII.3.2.3.3. Data evolution along the simulation for a particular pixel

As it has been said in section VII.3.2.3.2, when the user clicks the right mouse button on the projected image, two options appear, the *Change Parameter* label and also the ***Parameter's Evolution*** label. This label provides two options, either to select a pixel interactively over the map using a cursor, or provide the latitude and longitude of the points required from file. If the ***Interactive Selection*** label is chosen, the window in Figure VII.68 is presented.

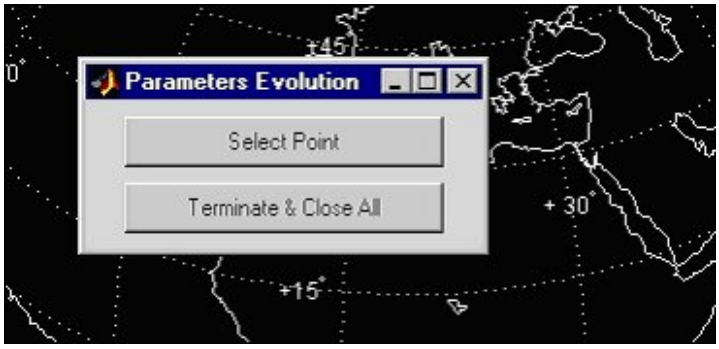


Figure VII.68. Interactive selection window

If the ***Select Point*** push button is chosen, a cursor appears to allow the user interactively select the point. If the ***Terminate & Close All*** button is pushed, the *Parameter's Evolution* tool is cancelled. If the selection is preferred, then after clicking the cursor on the required pixel Figure VII.69 opens.

This figure shows the evolution of the value of the data over the pixel, for all the snapshots containing it, and for all the parameters available through the cartography module. Each type of data can be individually disabled by clicking on the corresponding legend lines besides the texts. The user may notice also that the latitude and longitude of the requested point are displayed in the plot's title.

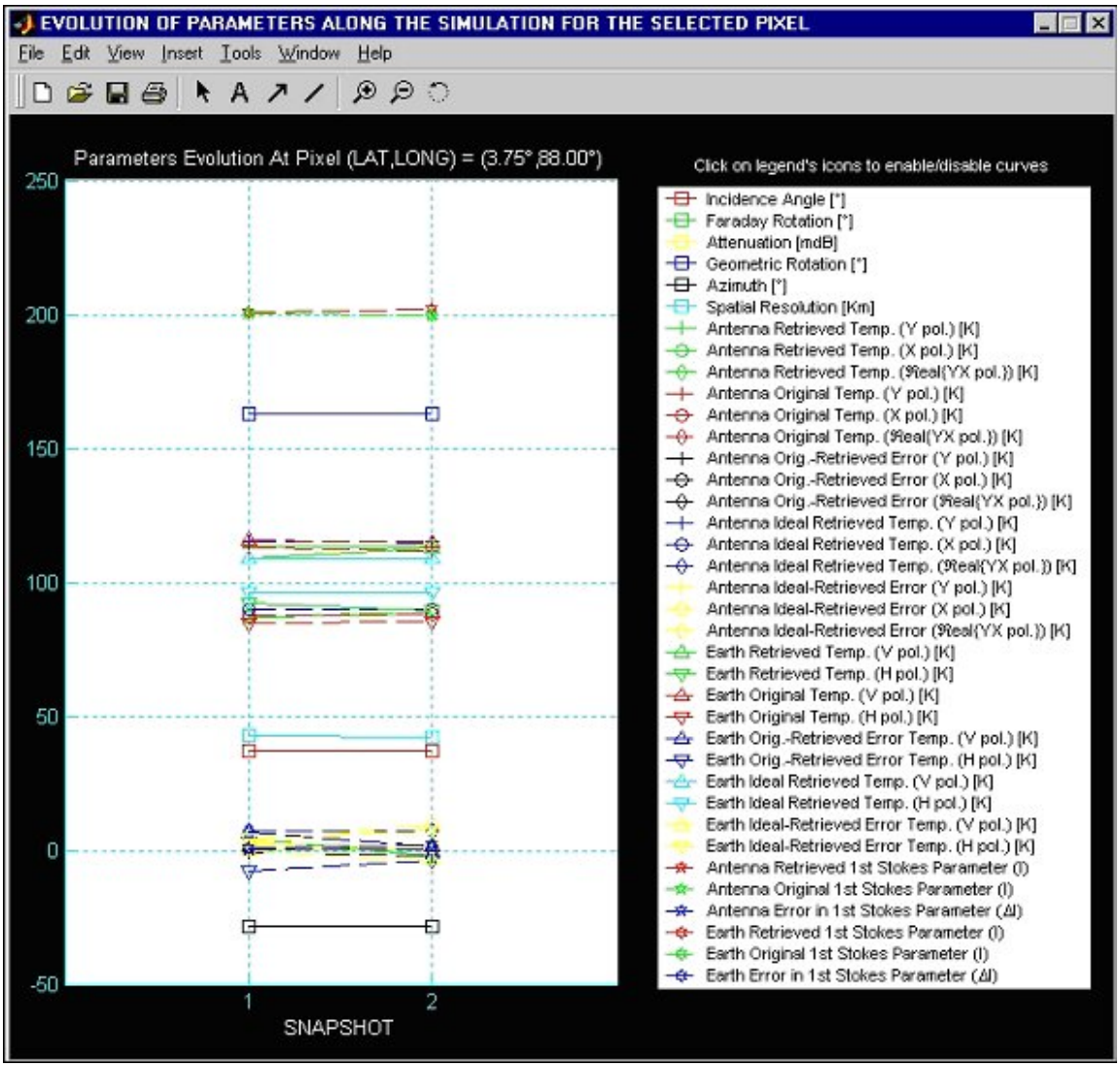


Figure VII.69. Parameter's evolution window, showing the evolution of all the data types over a selected pixel, for all the snapshots of the simulation that contain that pixel

If the ***Show Points From File*** label is selected, a dialog box like the one in Figure VII.70 is opened in order the user can select the file containing the latitudes and longitudes. Figures like VII.69 are then plotted in sequence.

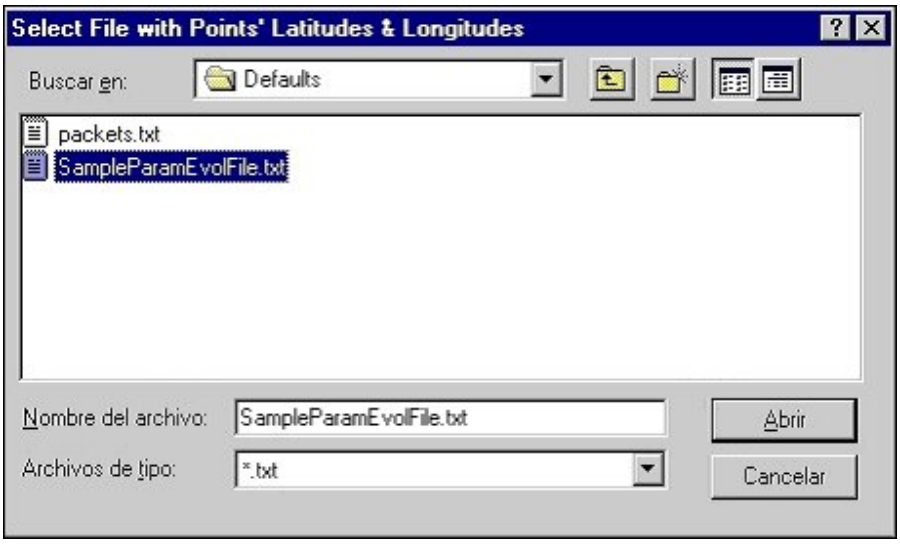


Figure VII.70. Dialog box opened to allow the user select the file that contains the pairs of latitude, longitude values for the points whose parameters' evolution is requested

The format needed for the file is one pair of geographic co-ordinates for each row - latitude first- in decimal degrees, separated by tabulators, and without any introductory text in the file:

-070.00	+062.00
+022.50	-063.00
+022.50	-062.75
+022.50	-062.50
+022.50	-062.25

SampleParamEvolFile.txt is a sample file containing some points and therefore showing the format needed for a provided user's file.

CHAPTER VIII. FIELD OF VIEW CALCULATOR

The *FOV Calculator* is intended to provide the user with a set of tools to evaluate the system's radiometric performances. These tools are provided to discuss the influence of the change of important parameters such as number of antennas, antenna temperature, etc..., on the instrument's radiometric performances in a statistical sense. This chapter reviews this set of tools, which are:

- the *Radiometric Accuracy* tool, to evaluate the system's error budget, whose use is reviewed in section VIII.1;
- the *Radiometric Sensitivity* tool, to compute this characteristics of the system's performance, which is explained in section VIII.2;
- the *Angular Resolution* tool, for evaluating the pixel resolution over the Earth's surface, the equivalent array factor and the synthesized half-power beamwidth, whose use is detailed in section VIII.3;
- the *Field Of View (FOV) Properties* tool, for plotting the radiometric accuracy and sensitivity, the incidence angle, the spatial resolution and the pixel's spot elongation, which is reviewed in section VIII.4;
- and the *Reliability* tool, to analyze the failure probability of the system, whose use is explained in section VIII.5.

If the user clicks on the ***FOV Calculator*** text label in the lower left box of the SEPS environment, an interface in the right box is presented, that consists of an upper area with 5 push buttons to call each of the *FOV Calculator* tools, and a box below for displaying each tool interface. Figure VIII.1 in section VIII.1.1 shows the default appearance of the calculator after clicking on the *FOV Calculator* text label.

The *FOV Calculator* in SEPS v4.0 can be run without having to execute a scenario simulation previously. This means not that dependence on the scenario parameters is avoided: some parameters as the antennas per arm, or the visibility taper window are obtained from the scenario definition section. But *FOV Calculator* tools can be used prior to that scenario simulation.

IMPORTANT ➔ *The user may notice that trying to manage other interfaces while executing any computation in SEPS forces the computations to abort. The user should wait until the end of one FOV Calculator tool's computations to continue using SEPS. Anyway, if the computation aborts for any reason, the user may ensure that the current folder is the* SEPS root *folder, in order to continue using the software without problems.*

For further information on these topics, the user may consult the AD/DD document (SO-TN-GMV-PLM-0003, section 6.58 and ANNEX C).

VIII.1. Radiometric Accuracy

The *Radiometric Accuracy* tool allows the user evaluating the system's radiometric error budget. To plot the interface where the parameters and the button to start computations is, the user may click on the ***Radiometric Accuracy*** push button in the *FOV Calculator* interface.

VIII.1.1. Input Parameters

The interface presented after calling this tool displays a number of input parameters, grouped by the type of radiometric error source. Three columns are available for entering parameters needed to compute the contribution of each error cause to the error budget:

- the first one requests the ***Initial Error Value*** of the required parameter;
- the second one allows the user to update the ***Drift / Time-Dependence***, or worst value in an orbit produced by a drift in temperature of 2ºC maximum. The units are K/([parameter units]·ºC);
- the third column allows to change the ***Sensitivity to Error*** for each error source. The units are K / [error source' units].

Figure VIII.1 shows the default appearance of the *Radiometric Accuracy* interface when clicking the corresponding button in the upper part of the *FOV Calculator*.

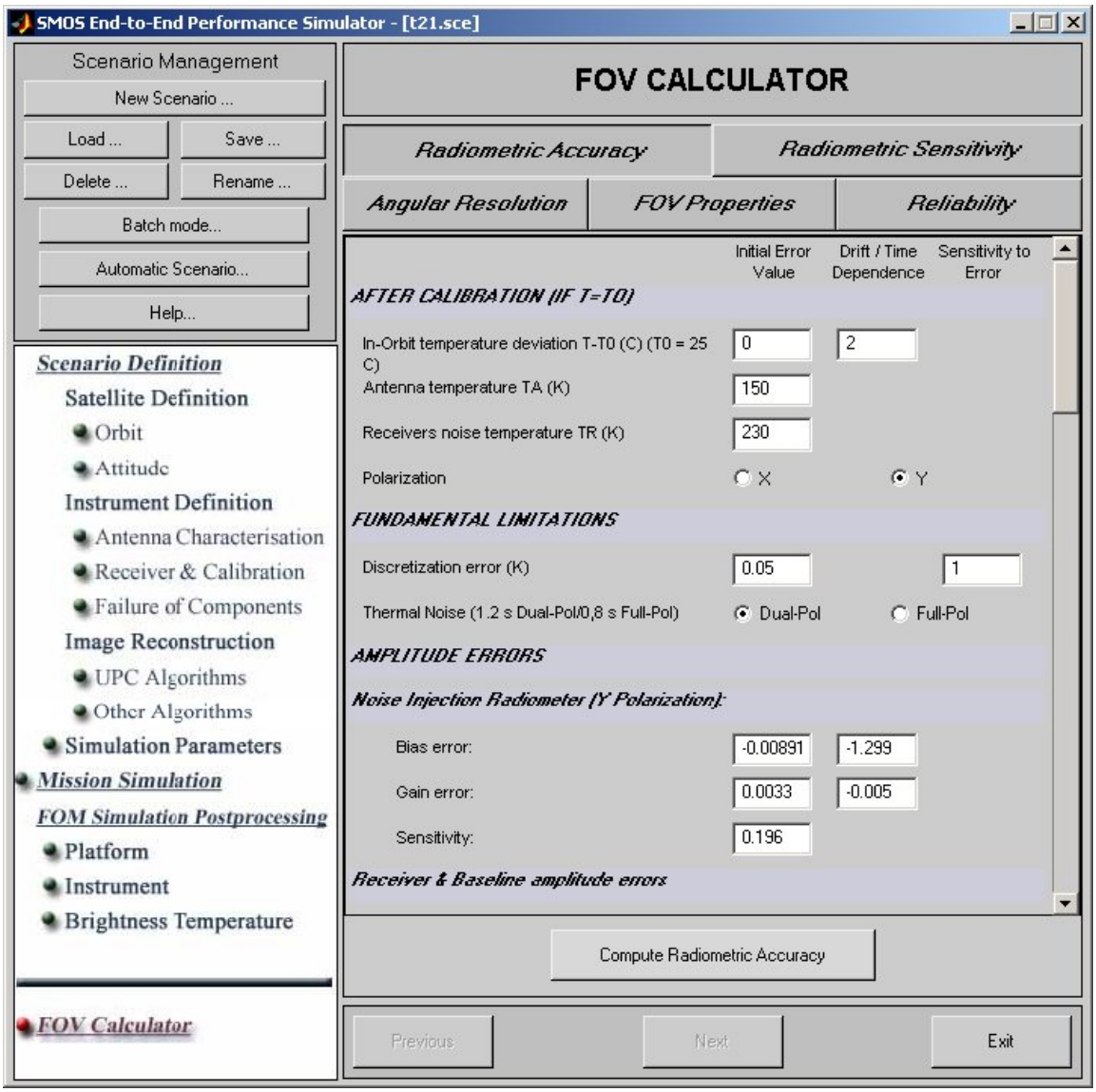


Figure VIII.1. Radiometric Accuracy *tool default interface for updating its input parameters.*

The user may notice that a slide bar at the right hand side of the interface is displayed, to move along the input parameters interface, as it can be seen in Figure VIII.2.

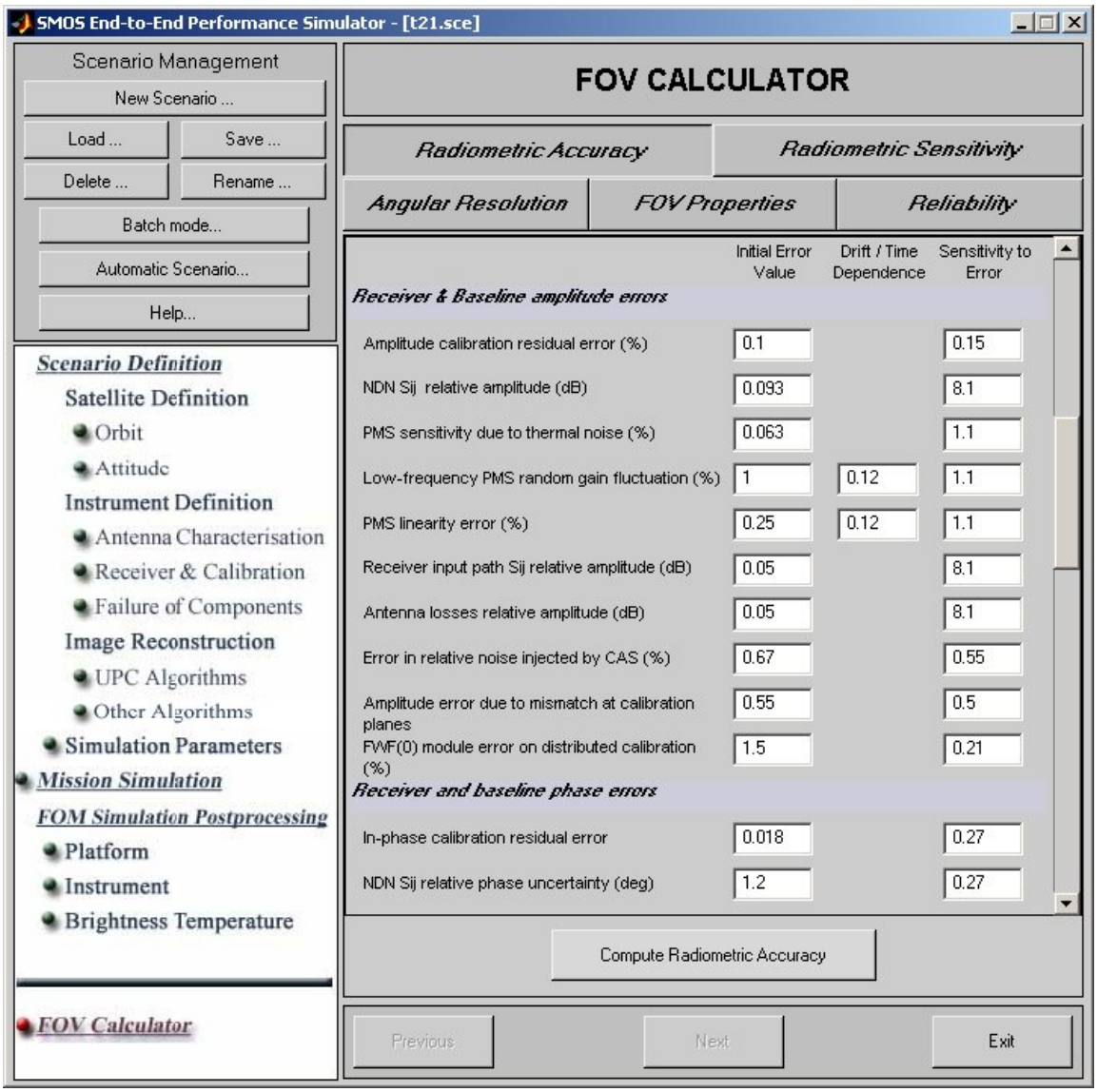


Figure VIII.2. Holding the left mouse button and moving it along the slide bar makes the rest of the input parameters appear.

In this section the input parameters needed to compute the system's radiometric accuracy at the boresight direction are briefly described. The user may consult the AD/DD document referred above (SO-TN-GMV-PLM-0003, ANNEX C, section d) or the SMOS System Performance Model and Error Budget document (reference SO-TN-UPC-PLM-0038 v2.1, 10 June 2005), to know more on these parameters.

The parameters that may be updated through this option before computing the radiometric accuracy are grouped by the type of error source. The list of parameters and a brief explanation follows.

First, the initial values for some parameters that are needed for the computations must be supplied:

- *In-orbit temperature deviation*: the user must supply the initial error value (by default this value is 0, that is, T=T0) and also the drift of this value with time.
- *Antenna temperature*: the average value of the measured scene.
- *Receiver's noise temperature*: in Kelvin (230 K by default).
- *Polarization*: *X*-antenna or *Y*-antenna.

Then, the contribution of each error source to the radiometric accuracy needs the following input parameters –grouped by error source type- to be updated by the user, if needed:

FUNDAMENTAL LIMITATIONS: are theoretical limitations of the Aperture Synthesis Concept.

- *Discretization error:* associated to the finite coverage of the (*u*,*v*) plane.
- *Thermal Noise.*

The above parameters are necessary to compute the radiometric sensitivity. The following ones affect the radiometric accuracy:

CHANNEL and CAS ERRORS: produce separable phase/amplitude errors, that is, they can be assigned to each particular channel. They can be calibrated by distributed noise-injection.

AMPLITUDE ERRORS

- *Noise Injection Radiometer*: the user may supply the values for the NIR characterization, which are

i. *Bias* (the initial value and drift).
ii. *Linearity* (the initial value and drift).
iii. *Radiometric Sensitivity*.

Receiver and Baseline Amplitude Errors:

- *Amplitude Calibration Residual error*: residual quadrature error after calibration, in %. The user may change its initial value and also the sensitivity to error.
- *NDN Sij relative amplitude*: relative amplitude of the noise distribution network scattering parameters, in dB. The user may change its initial value and also the sensitivity to error.
- *PMS sensitivity due to thermal noise*: error due to *Power Measurement System* finite integration time, in %. The user may change its initial value and also the sensitivity to error.
- *Low-frequency PMS random gain fluctuation*: the receiver's gain fluctuation, in %. The user may change its initial value, its drift value and also the sensitivity to error.
- *PMS linearity error*: ratio of the actual slope of the detector (mV/K) and the measured one, in %. Note: second order effects have a minor impact in the RMS radiometric accuracy (pixel bias) and are not included in this table. In SEPS, these parameters are entered through data files.
- *Receiver input path Sij relative amplitude*: measured between antenna plane and the TRFOP connector for every switch, in dB. The user may change its initial value and also the sensitivity to error.
- *Antenna losses relative amplitude*: the mismatch of the antenna losses, inducing noise included in the receiver's noise temperature and affecting the denormalization of the visibility samples. The user may change its initial value and also the sensitivity to error.
- *Error in relative noise injected by CAS*: in %. The user may change its initial value and also the sensitivity to error.
- *Amplitude error due to mismatch at calibration planes*: measured at reference planes and not corrected, in %. The user may change its initial value and also the sensitivity to error.

- *FWF(0) module error on distributed calibration*: associated to the pair of receivers involved in the baseline, measured in %. The user may change its initial value and the sensitivity to error.

PHASE ERRORS

- *In-phased calibration residual error*: with an integration time of 1.2 seconds. The user may change its initial value and also the sensitivity to error.
- *NDN Sij relative phase uncertainty*: relative phase of Sij of the noise distribution network relative to the NIR path, in sexagesimal degrees. The user may change its initial value and also the sensitivity to error.
- *Receiver input path Sij relative phase uncertainty*: relative phase of the polarization switch, measured between antenna plane and TRF connector, in sexagesimal degrees. The user may change its initial value and also the sensitivity to error.
- *Path antenna plane to antenna geometric center*: the phase in the path from the phase center and the antenna reference plane, in sexagesimal degrees. The user may change its initial value and also the sensitivity to error.
- *Residual quadrature error*: internal calibration quadrature error, in sexagesimal degrees. The user may change its initial value and also the sensitivity to error.
- *Phase error due to mismatch at calibration planes*: measured at reference planes and corrected, in %. The user may change its initial value and also the sensitivity to error.
- *In-band frequency dependent quadrature error*: the frequency dependence of quadrature errors, in sexagesimal degrees.
- *FWF(0) phase error in distributed calibration*: associated to the pair of receivers involved in the baseline, measured in sexagesimal degrees. The user may change its initial value and the sensitivity to error.

OTHER SOURCES OF ERROR:

- *Sampling skew error*: bias error in sampling, measured in ns. The initial value is needed.
- *Sampling jitter error*: random error in sampling. Measured in ns. The initial value is needed.

ANTENNA ERRORS: affect the exploration of the scene, and they must be accurately characterized on ground.

- *Antenna voltage pattern phase ripple*: this ripple is the measurement error, not the rms mismatch between antenna patterns themselves. The user must supply the initial value for this parameter and also its sensitivity to error.
- *Antenna voltage pattern amplitude ripple*: this ripple is the measurement error, not the rms mismatch between antenna patterns themselves. The user must supply the initial value for this parameter and also its sensitivity to error.
- *Antenna XP*: the antenna finite cross-polarization ratio in dB. The default value is computed as average of antenna measurements in alias–free FOV.
- *Switch isolation*: the value of the isolation in the polarization switch in dB. The user must supply the initial value and the drift.
- *Geometric position uncertainty (x-y)*: antenna phase center measurements uncertainty in-plane (array x-y plane) contribution to the radiometric accuracy needs its initial value and the sensitivity to error.
- *Geometric position uncertainty (z)*: antenna phase center measurements uncertainty off-plane (array z plane) contribution to the radiometric accuracy needs its initial value and the sensitivity to error.
- *Array arm thermo-elastic deformation in-plane*: array arm oscillations in the x-y plane measured or known, may be updated by the user in sexagesimal degrees. Also the drift, the sensitivity to error and the sensitivity to error with the antenna temperature T_A and T_o may be updated.
- *Array arm thermo-elastic deformations off-plane*: array arm oscillations in the z plane measured or known, may be updated by the user in sexagesimal degrees. Also the drift and the sensitivity to error may be updated.
- *Hub arm thermo-elastic deformation in-plane*: hub arm oscillations in the x-y plane, may be updated by the user in sexagesimal degrees. Also the drift and the sensitivity to error may be updated.
- *Hub arm thermo-elastic deformation off-plane*: hub arm oscillations in the z plane may be updated by the user in sexagesimal degrees. Also the drift and the sensitivity to error may be updated.

Antenna rotation: antenna's polarization axis' alignment error, needs the initial value to be set.

- *Comparator threshold and U-noise injection correction*: residual offset error in each correlator after using the 1-0 count technique, measured in cu. The user may change its initial value and the time dependence.
- *SELF-RFI and Flat Target Response correction*: the self-radiofrequency interference can induce additional residual correlations appearing as extra offsets. Measured in cu. The user may change its initial value.

OTHERS: Sun etc.

Other external error sources –such as Sun interference- can be taken into account despite there is no tool to evaluate its impact on the instrumental radiometric error sources. A box is available to introduce the estimated impact of the radiometric accuracy.

VIII.1.2. Outputs

The radiometric accuracy tool allows the user to evaluate this fundamental parameter from the values supplied in the *Radiometric Accuracy* section. By pressing the **Compute Radiometric Accuracy** button the user may compute and display the results shown in the following images:

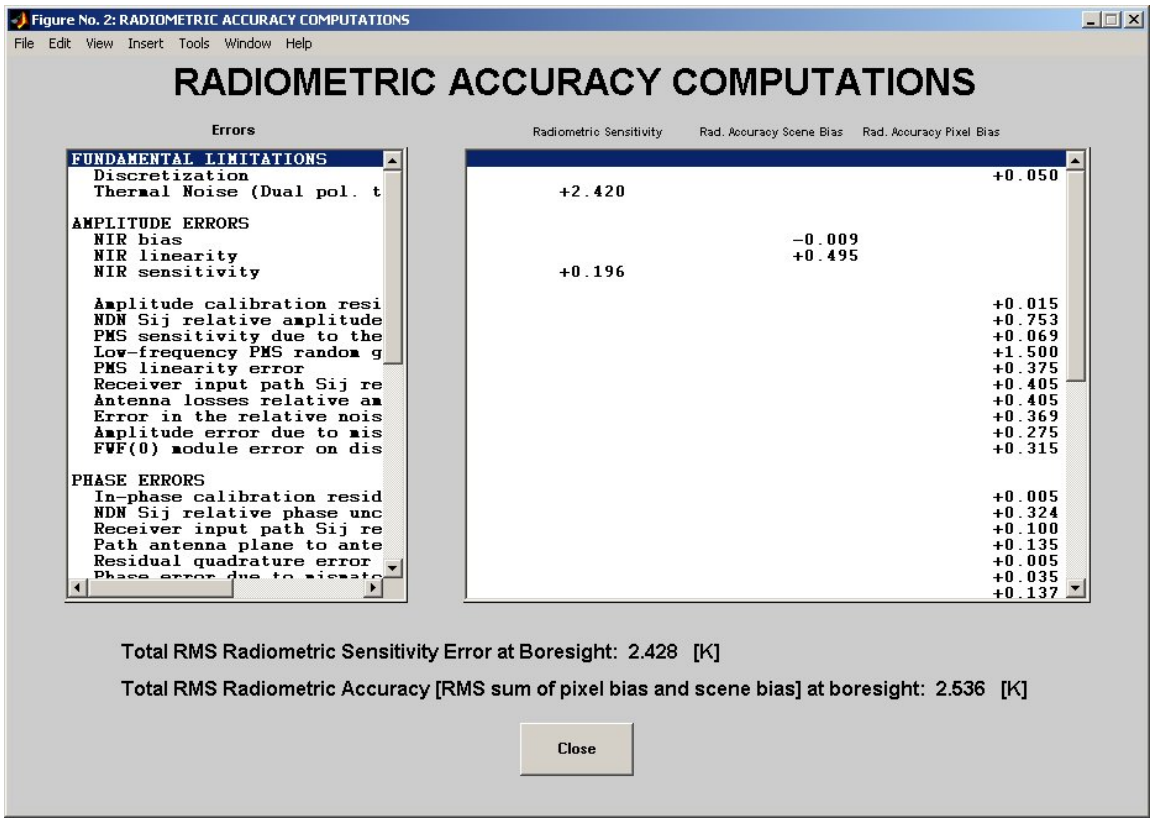


Figure VIII.3. Radiometric Accuracy *tool default interface for updating its input parameters.*

Two list boxes are used in Figure VIII.3 to display the computed values of the individual contributions to the calculation of the radiometric accuracy. The left list box shows the names of the error sources (tabulated text lines), grouped by the type of error source (non-tabulated text lines). The right list box contains the resulting values computed for each error source, and organized in three columns, depending on the output parameter that they are affecting to:

- ***Radiometric Sensitivity***: affected by the thermal noise and the Noise Injection Radiometer Sensitivity;
- ***Radiometric Accuracy Scene Bias***: includes the contribution of the Noise Injection Radiometer bias and linearity;
- ***Radiometric Accuracy Pixel Bias***: is affected by all the other error sources.

The user may notice that both left and right list boxes are connected. Selecting one parameter in the left box highlights the corresponding computed values in the right box. And clicking over a number in the right box highlights the error source name in the left box.

Finally, both the total radiometric sensitivity - computed using the values from the Error Budget (see document SO-TN-UPC-PLM-0002 rev. 3.20)- and the total radiometric accuracy values at boresight direction are presented at the bottom of the figure. A ***Close*** push button is available for terminating the user of the *Radiometric Accuracy* tool.

VIII.2. Radiometric Sensitivity

The Radiometric Sensitivity tool allows the user evaluating this system's characteristic. Some computations are done in SEPS v4.0 that had no to be executed in Release 1, that have been included in order to assure independence of the *FOV Calculator* from the rest of the simulator –or at least, to allow executing its tools without having to execute a simulation previously.

VIII.2.1. Input Parameters

The parameters needed to compute the *Radiometric Sensitivity* may be updated by the user through the interface displayed after clicking ***the Radiometric Sensitivity*** push button in the upper part of the *FOV Calculator* interface. The interface can be seen in Figure VIII.4 and these parameters are:

- *Antenna Temperature*: units are Kelvin.
- *Receiver's bandwidth*: the narrow bandwidth obtained from the cascade of bandwidths of the receiver's blocks. Units are MHz.
- *Integration time*: the period of time along which the integration of a snapshot is done. Units are seconds.

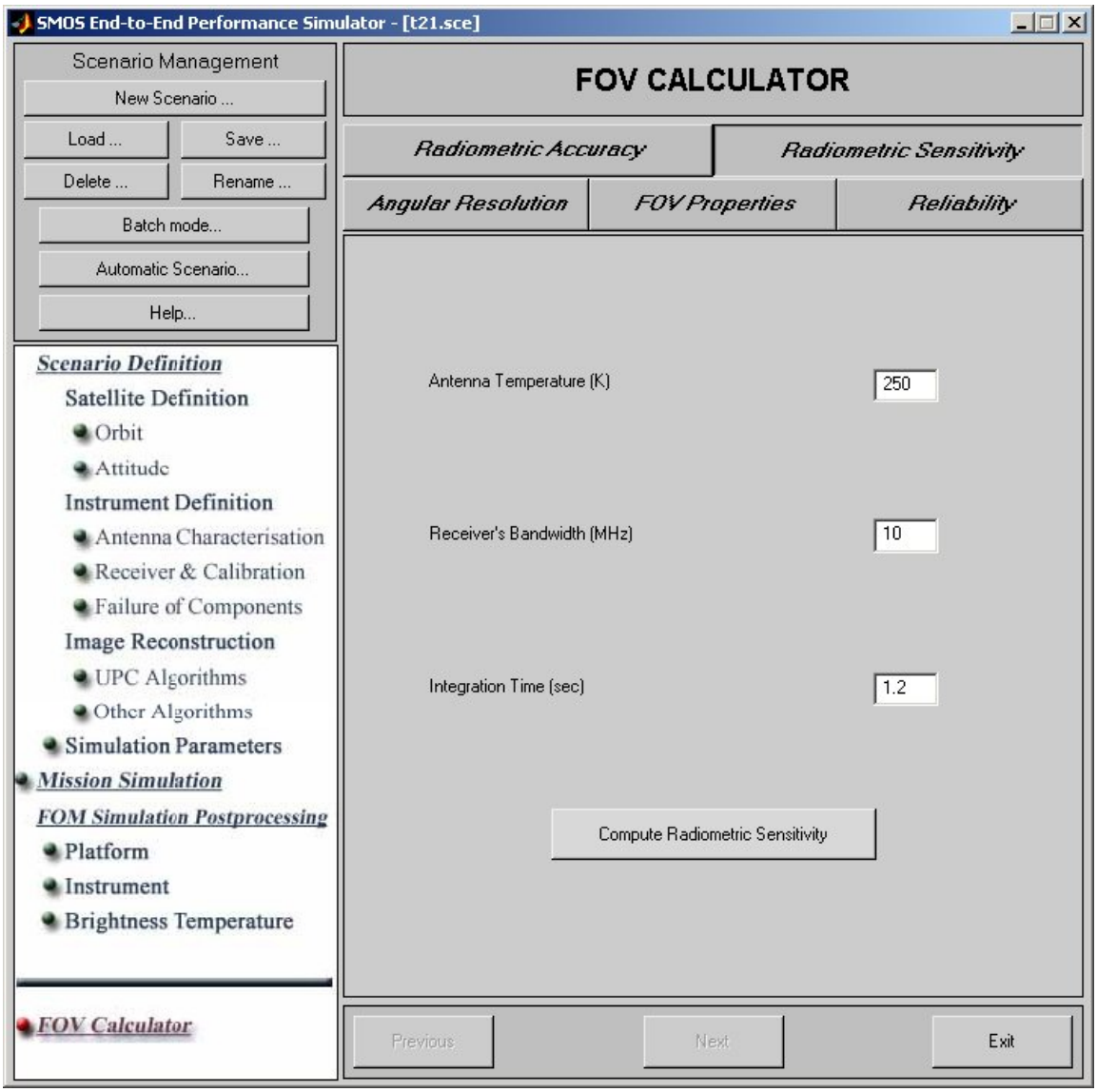


Figure VIII.4. Radiometric Sensitivity *tool interface for updating its input parameters, and launching the computation through the push button in the lower part.*

VIII.2.2. Outputs

To compute the radiometric sensitivity the user may press the ***Compute Radiometric Sensitivity*** button in the lower part of the interface. Figure VIII.5 shows the window that is opened displaying the computed radiometric sensitivity at boresight. Two values are computed, using the redundancy averaging capability or not. Enhancement

introduced by redundancy can be assessed through this tool. The user may press the ***OK*** push button to close the window.

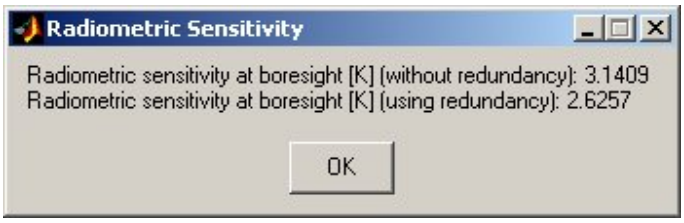


Figure VIII.5. Radiometric Sensitivity *output figure.*

The user may notice that these values can be different from the ones computed using the *Radiometric Accuracy* tool. The *Radiometric Accuracy* tool uses the values measured for the Error Budget document (see reference SO-TN-UPC-PLM-0007 rev. 7.1), while the *Radiometric Sensitivity* tool uses the values from the *Scenario Definition* interfaces and also the tool's inputs (see previous section VIII.2.1) which may be different, as in example the receiver's bandwidth.

VIII.3. Angular Resolution

The Angular Resolution tool is provided to allow computing the SMOS/MIRAS field of view shape, and to interactively evaluate the angular resolution and elongation of a user-selected pixel, over the Earth's surface and also in the director cosines domain.

VIII.3.1. Input Parameters

The user may click the ***Angular Resolution*** button in the upper zone of the *FOV Calculator* interface, to show the input parameters list necessary for computing this tool, as seen in Figure VIII.6. These input parameters are:

- *Steering angle*: has a main influence in the orientation of the SMOS FOV. It can be either 0° or 30°. The user may consult Chapter I for knowing more on the array's orientation.

- *Method for computing the synthesized half-power beamwidth*: the interface requests whether to use an exact or an approximate method to compute the half-power synthesized beamwidth, being the approximate method quicker, but slightly less accurate.
- *Include fringe-wash effects*: fringe-wash effect (radial blurring of the equivalent array factor or impulse response) in the computation of the pixel's spot shape.
- *Calculate and show equivalent array factor* (impulse response): the equivalent factor for the current array configuration can be computed and shown if requested by the user. If so, a waitbar is displayed informing the user about the progress of the equivalent array factor computation.
- *Co-ordinates for equivalent array factor center*: by default they are (0,0), but the user has the possibility to change it.

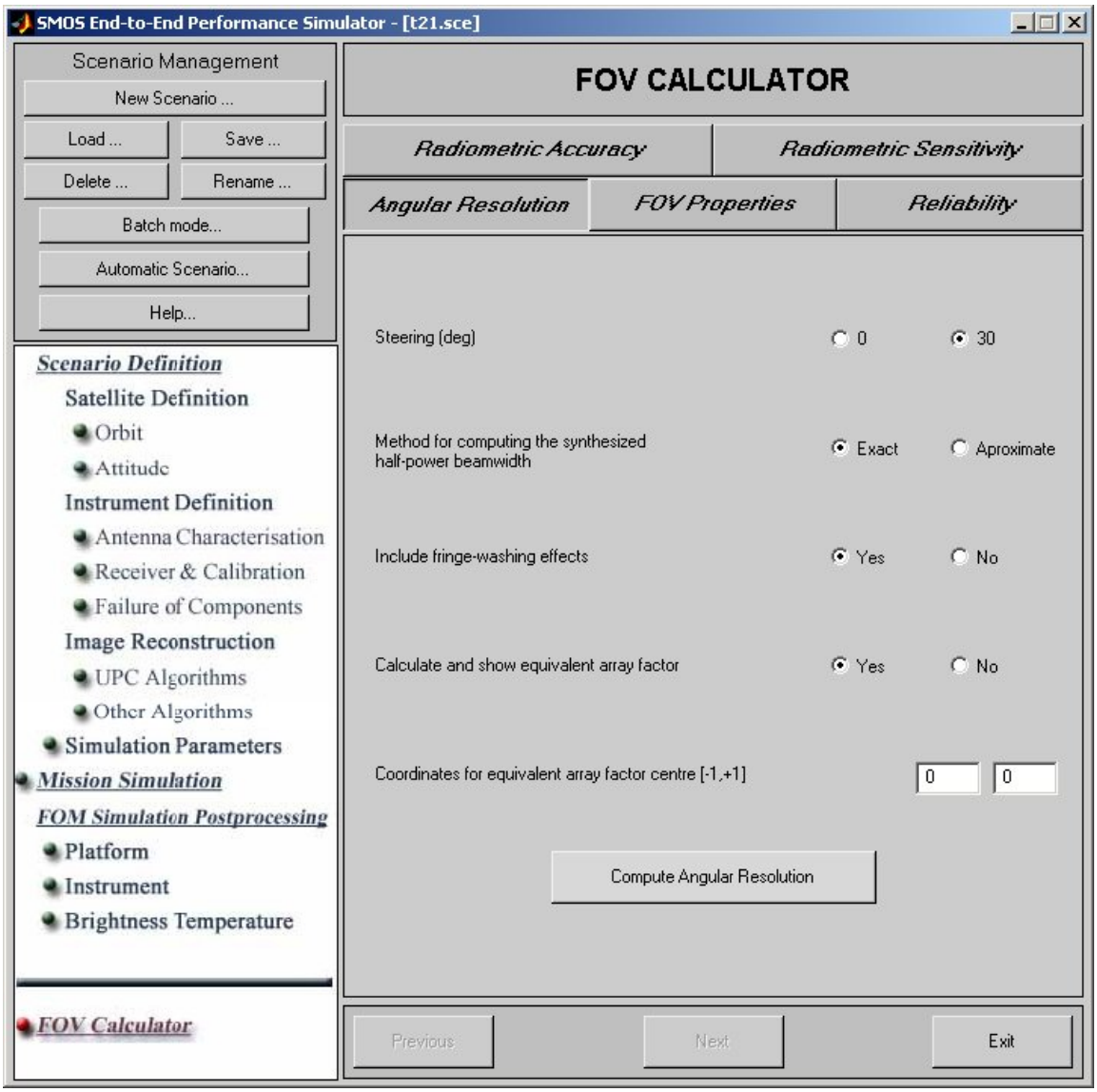


Figure VIII.6. Angular Resolution *tool interface for updating its input parameters, and launching the computation through the push button in the lower part.*

VIII.3.2. Outputs

Once the input parameters are updated, the user may click on the ***Compute Angular Resolution*** push button to start computations. Some waitbar figures are presented to allow the user know the progress of computations. After finishing them, two, three or four figures are presented, depending on the user requirements about the

equivalent array factor computation, and also on the platform's pointing –this is explained later in this section.

If the computation of the equivalent array factor is requested, Figure VIII.7 is displayed, centered at the (ξ,η) co-ordinates updated through the input parameters interface. The plot's title specifies the window used, as is set in the *UPC Algorithms* interface of the *Image Reconstruction* section in the *Scenario Definition* box (by default, the taper window is Blackmann).

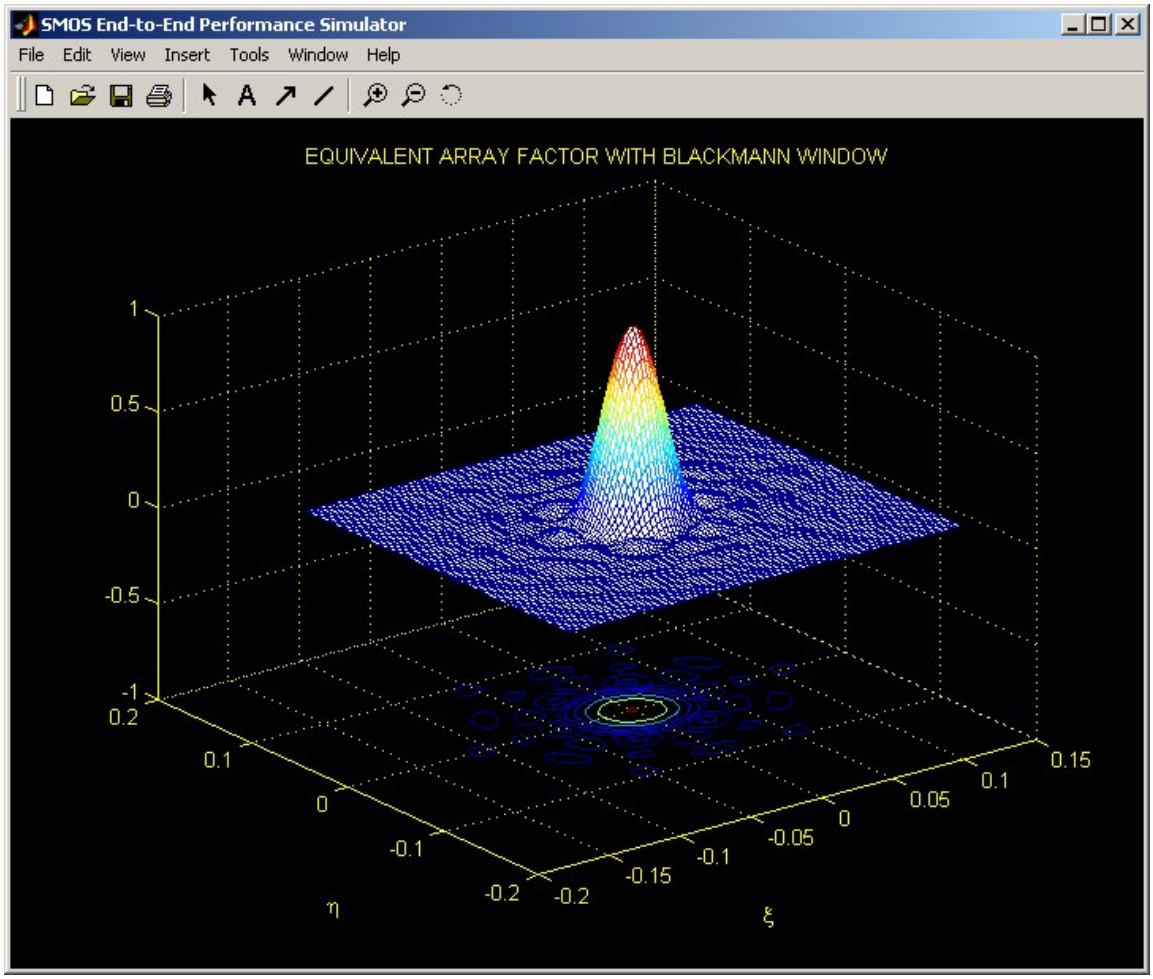


Figure VIII.7. Equivalent array factor centered at the co-ordinates requested by the user -(0,0) by default-. *The plot's title specifies the type of window used to taper visibilities.*

Once the equivalent array factor is displayed, the tool opens two figures where the SMOS field-of-view (FOV) is drawn in director cosines domain co-ordinates and in Earth along-track/cross-track co-ordinates. The first figure is shown below.

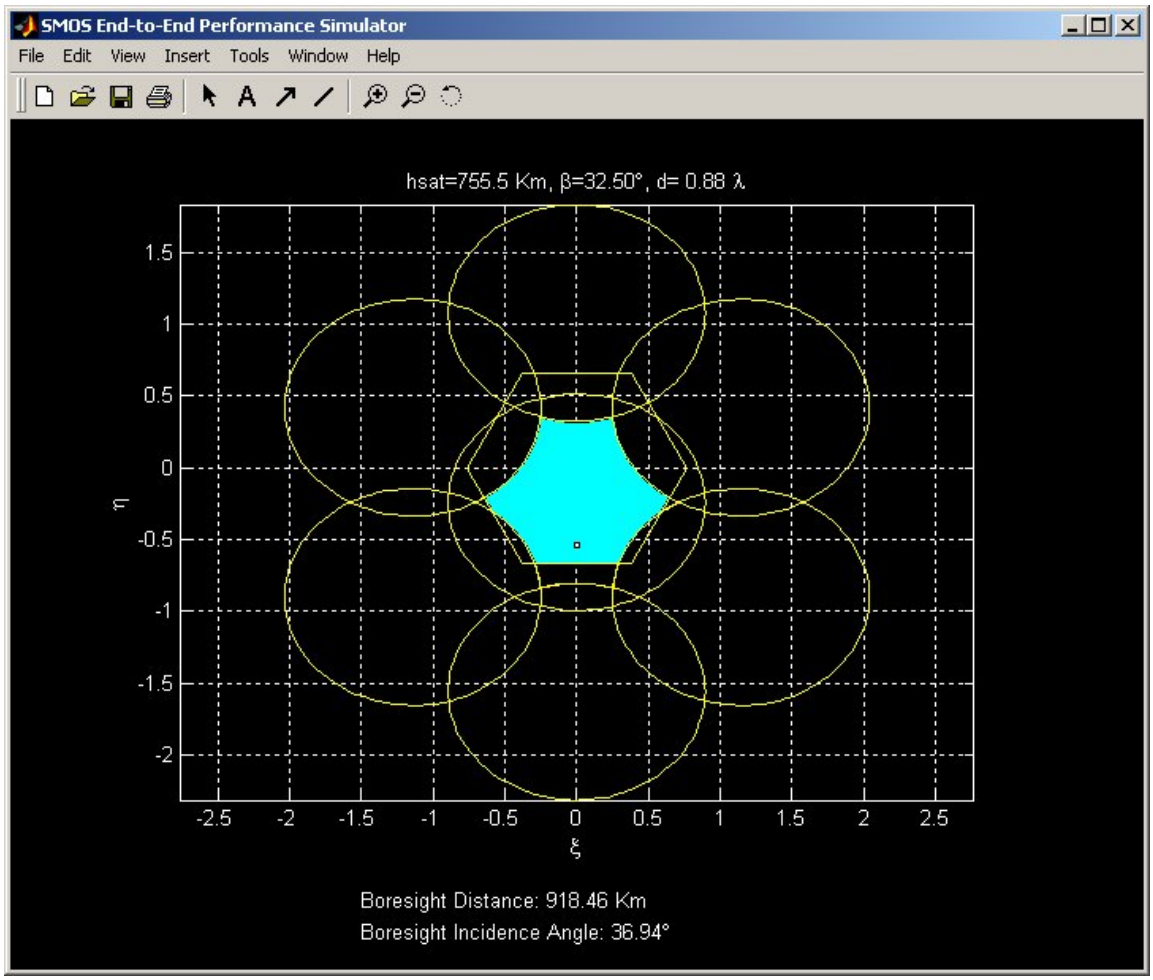


Figure VIII.8. Angular Resolution *in the director cosines domain, showing the field of view, the Earth's aliases and the hexagon. The inner cyan area is the SMOS/MIRAS field of view. The white spots represent the pixels selected by the user to represent their spatial resolution and elongation.*

The above image plots the Earth shape in (ξ,η) co-ordinates, its aliases and the hexagon, and fills the inner shape corresponding to the SMOS FOV in cyan color. Also a white spot representing the subsatellite point appears. The most important input parameters are displayed in the plot's title, while the boresight distance and incidence angle is specified below the plot.

The second figure, Figure VIII.9, plots the FOV in earth co-ordinates. The satellite across-track direction is plotted in the abscissa axis and the satellite along-track direction in the ordinate direction, as shown below. Also the constant incidence angles' curves are plotted. The most important parameters are specified in the plot's title

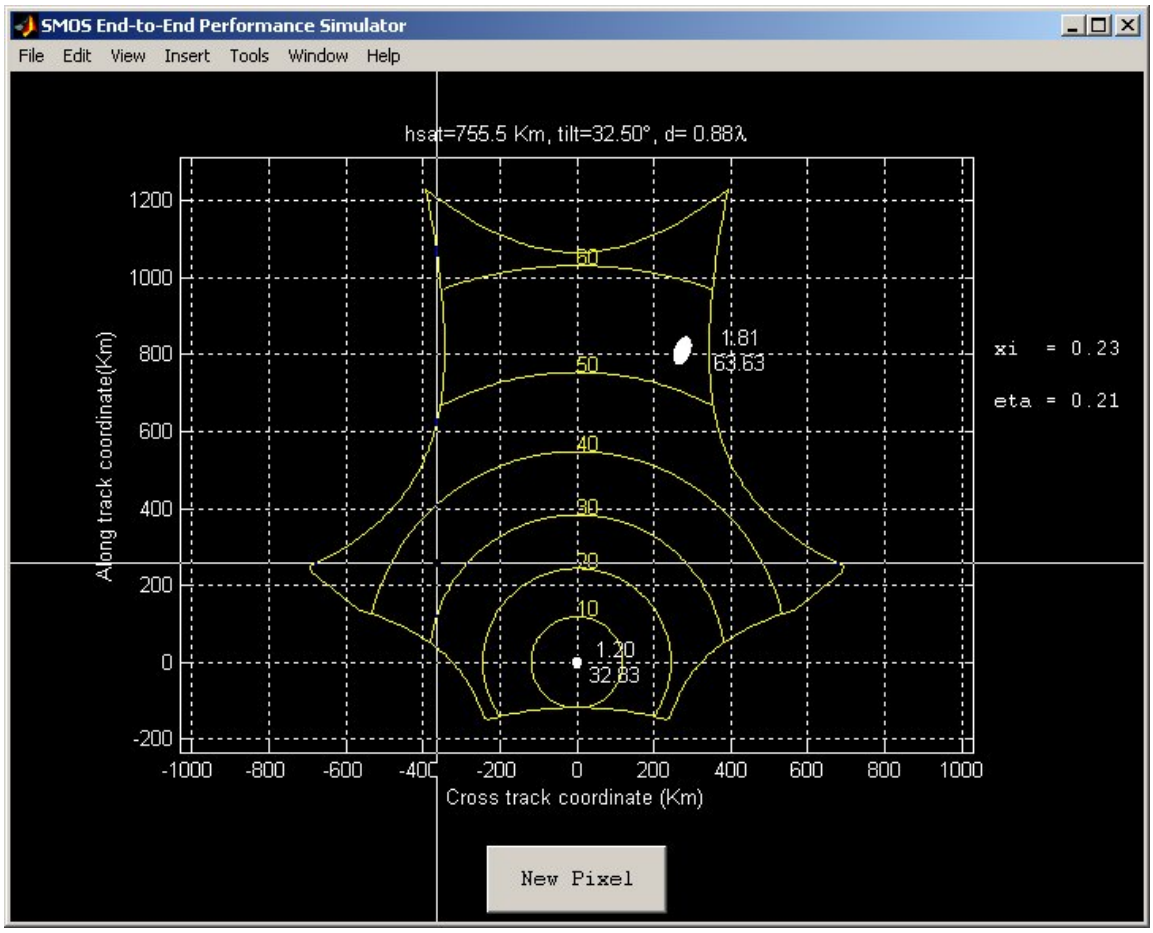


Figure VIII.9. Angular Resolution *over the Earth's surface, showing the field of view contour and also the constant incidence angles circles. The white ellipsoidal spots represent the pixels selected by the user to represent their spatial resolution and elongation.*

In the figure above, the user may press the ***New Pixel*** button to start a pixel selection using a cursor. Then, the spot for the selected point is computed and drawn. The spatial resolution and elongation of the spot in that particular position of the FOV are computed. These two parameters are displayed both in the figure, next to the white ellipse

representing the pixel's spot, and the *MATLAB Command Window*. In the figure the elongation is the upper displayed number, and the lower number is the spatial resolution in kilometres. In the *MATLAB Command Window* these parameters are displayed as follows

>> a/b = 1.80 sqrt(a*b) = 63.33

where the first number is the elongation and the second is the spatial resolution. The tool also provides the (ξ,η) co-ordinates for the selected points in the right hand side of the figure. As it can be observed in the cosines domain image shown above, also the pixel's spot is plotted in that figure.

Finally, the last figure presented is a small window displaying the computed value for the synthesized half-power beamwidth. The user may press the ***OK*** button to finish it.

Figure VIII.10. The fourth results figure of the Angular Resolution *tool displays the computed value for the synthesized half-power beamwidth.*

As it has been said above, occasionally the figures presented can be only two. This occurs when the equivalent array factor is no required and when the platform is pointing towards the sky and therefore it has not the Earth's shape in its scope. Then, the figure over the Earth's surface is not presented, and therefore the pixel selection is not available. In the figure plotting the FOV in the director cosines domain, the FOV is limited then by the unit circles of the alias images. Also the hexagon in white color is plotted, although in this case is not limiting the FOV. Figure VIII.11 displays the resulting figure.

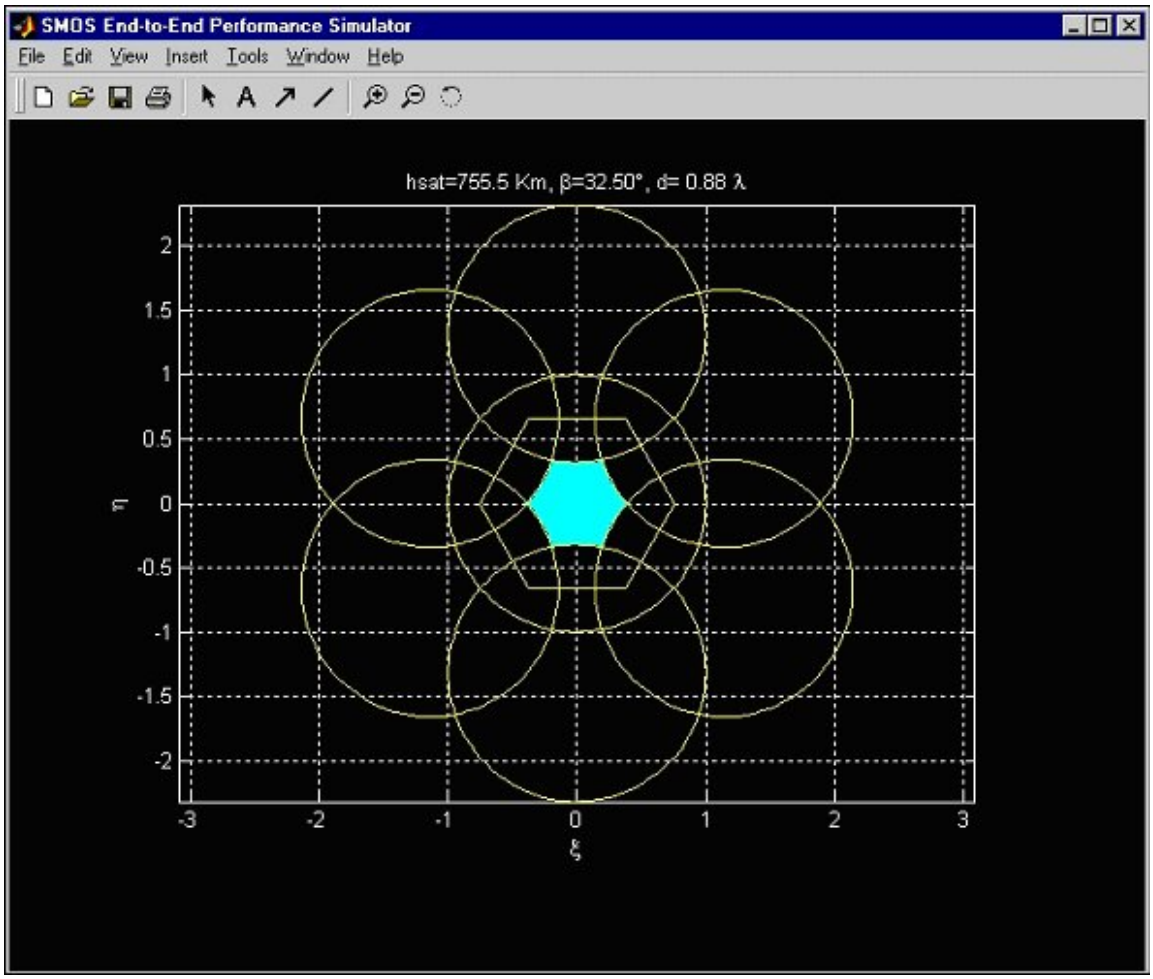


Figure VIII.11. Angular Resolution *in the director cosines domain when the platform is pointing towards the sky.*

VIII.4. FOV Properties

The Field Of View (FOV) Properties tool is provided to allow evaluating the system's radiometric performances –accuracy and sensitivity-, the incidence angle, and the spatial resolution and elongation, within the area comprehended in the SMOS/MIRAS field of view.

VIII.4.1. Input Parameters

The user may click the FOV Properties button in the upper zone of the FOV Calculator interface, to show the input parameters list necessary for computing this tool, as seen in Figure VIII.12. These input parameters are:

- *Steering angle*: has a main influence in the orientation of the SMOS FOV. It can be either 0° or 30°. The user may consult Chapter I to know more on the array's orientation.
- *Method for computing the synthesized half-power beamwidth*: the interface requests whether to use an exact or an approximate method to compute the half-power synthesized beamwidth, being the approximate method quicker, but slightly less accurate.
- *Include fringe-wash effects*: fringe-wash effect (radial blurring of the equivalent array factor or impulse response) in the computation of the pixel's spot shape.
- *Antenna temperature*: needed for the computation of the radiometric sensitivity curves. Units are Kelvin.
- *Receiver's bandwidth*: the narrow bandwidth obtained from the cascade of bandwidths of the receiver's blocks. Units are MHz.
- *Integration time*: the period of time along which the integration of a snapshot is done. Units are seconds.

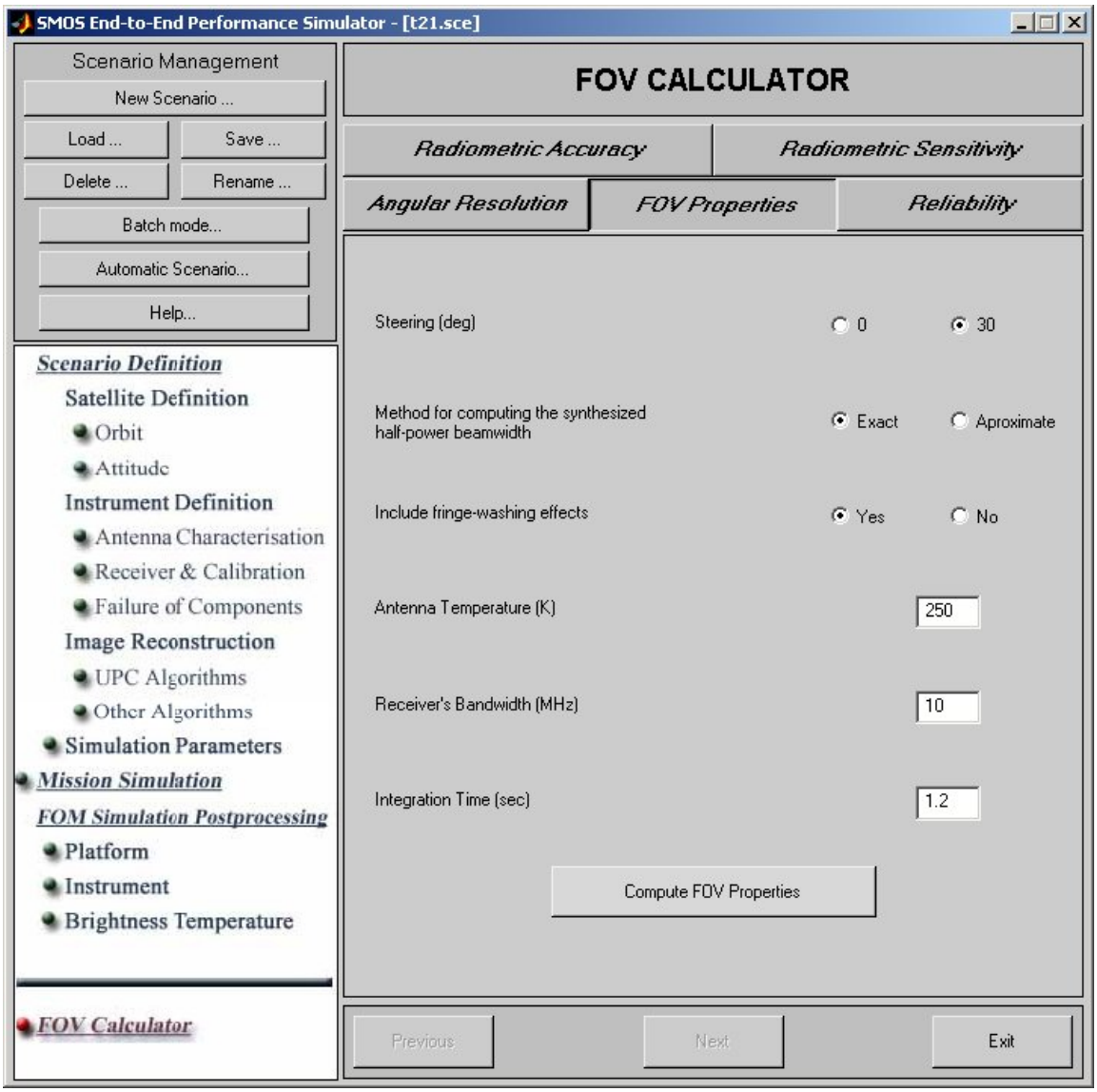


Figure VIII.12. FOV Properties *tool interface for updating its input parameters, and launching the computation through the push button in the lower part.*

VIII.4.2. Outputs

Once the input parameters are updated, the user may click on the ***Compute FOV Properties*** push button to start computations. Some waitbar figures are presented to allow the user know the progress of computations. After finishing the computations, two or three figures are presented showing the results.

The main difference for this *FOV Calculator* tool between the SEPS v1.0 and Release 4, is that this one shows now the computed system's performances in both the director cosines domain and over the Earth's surface. This is the reason why the presented figures can be either two or three –depending on the platform's pointing, the figure displaying the properties over the Earth's surface is not presented-.

First figure shown is the one showing iso-curves in the director cosines domain. An example can be seen in Figure VIII.13.

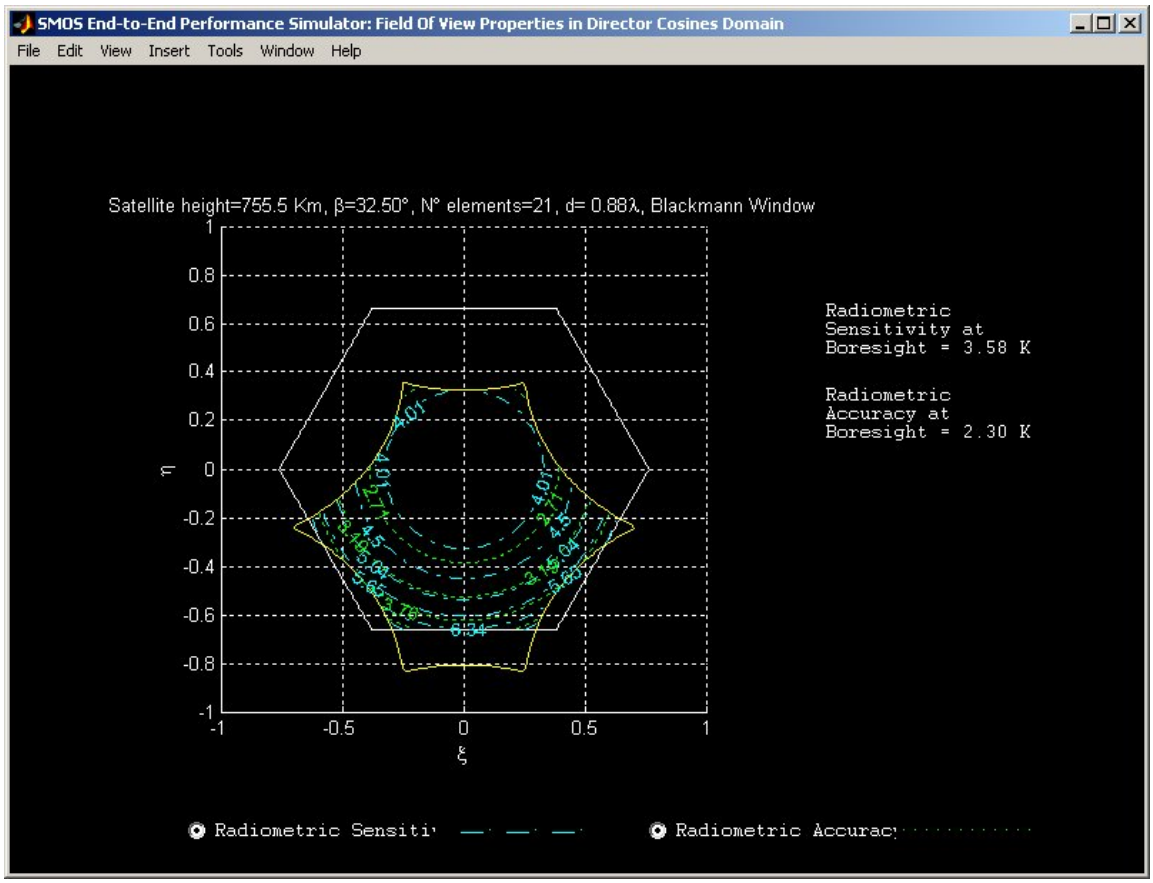


Figure VIII.13. FOV Properties *tool results figure in the director cosines domain.*

The figure plots the SMOS field-of-view limited by the Earth's shape aliases in yellow color, and also in white color the hexagon. For the points inside the FOV, the figure plots the curves of iso-radiometric sensitivity and iso-radiometric accuracy –it is nonsense

that other parameters such as spatial resolution and elongation are plotted in the director cosines domain-. The legend at the bottom of the figure allows the user whether displays or not the curves by clicking the mouse on the radio buttons identifying each type of curves. The most important parameters needed to define the system –such as tilt angle, or the window used to taper the visibilities- are included in the plot's title. And besides the plot, the radiometric sensitivity and accuracy at boresight direction are displayed.

The second figure plots all the curves over the Earth' surface, including also the iso-spatial resolution, iso-elongation, and iso-incidence angle, apart from the iso-radiometric sensitivity and iso-radiometric accuracy. The system's most important parameters are also included in the plot's title, and besides the plot the radiometric characteristics at the boresight direction are displayed.

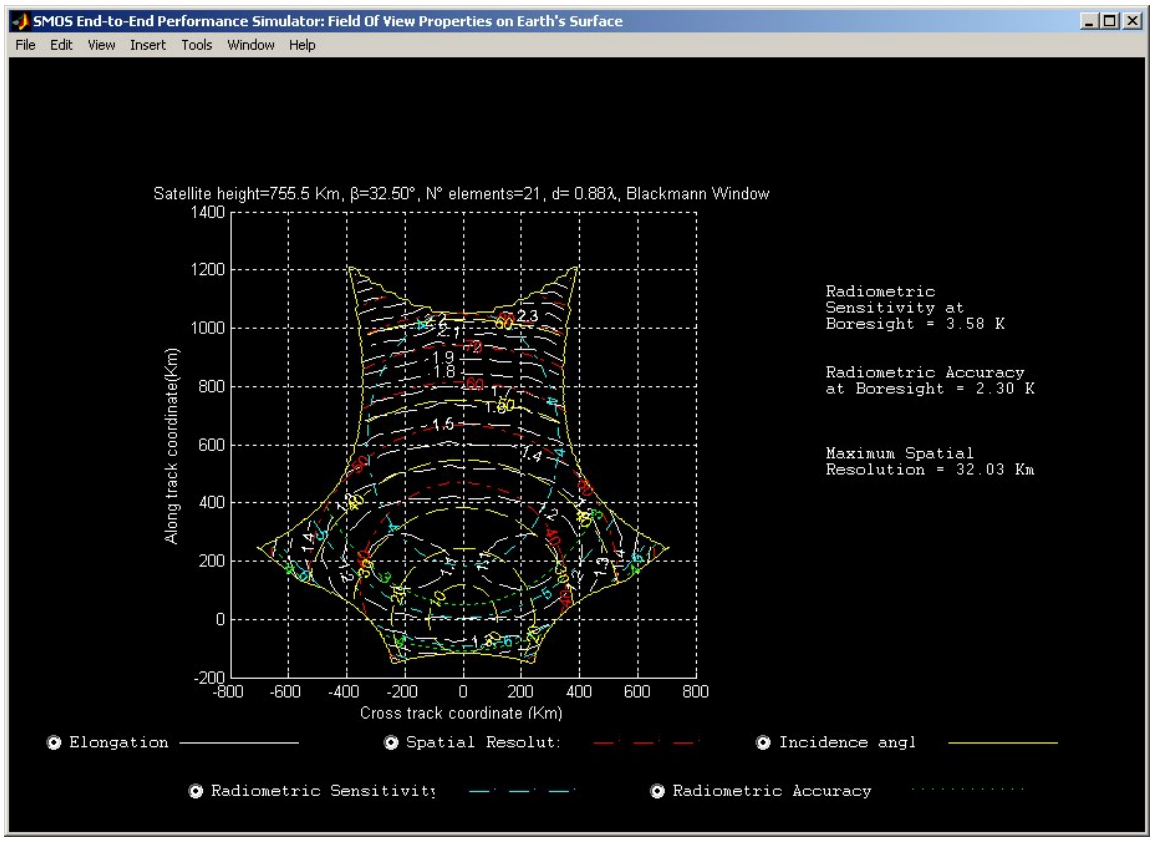


Figure VIII.14. FOV Properties *tool results figure over the Earth's surface.*

The legend at the bottom of the figure allows the user whether displays or not the curves by clicking the mouse on the radio buttons identifying each type of curves. In Figure VIII.15 it can be seen an example of the window's aspect after disabling some parameter's curves –in this case, the spatial resolution, the incidence angle and the radiometric accuracy.

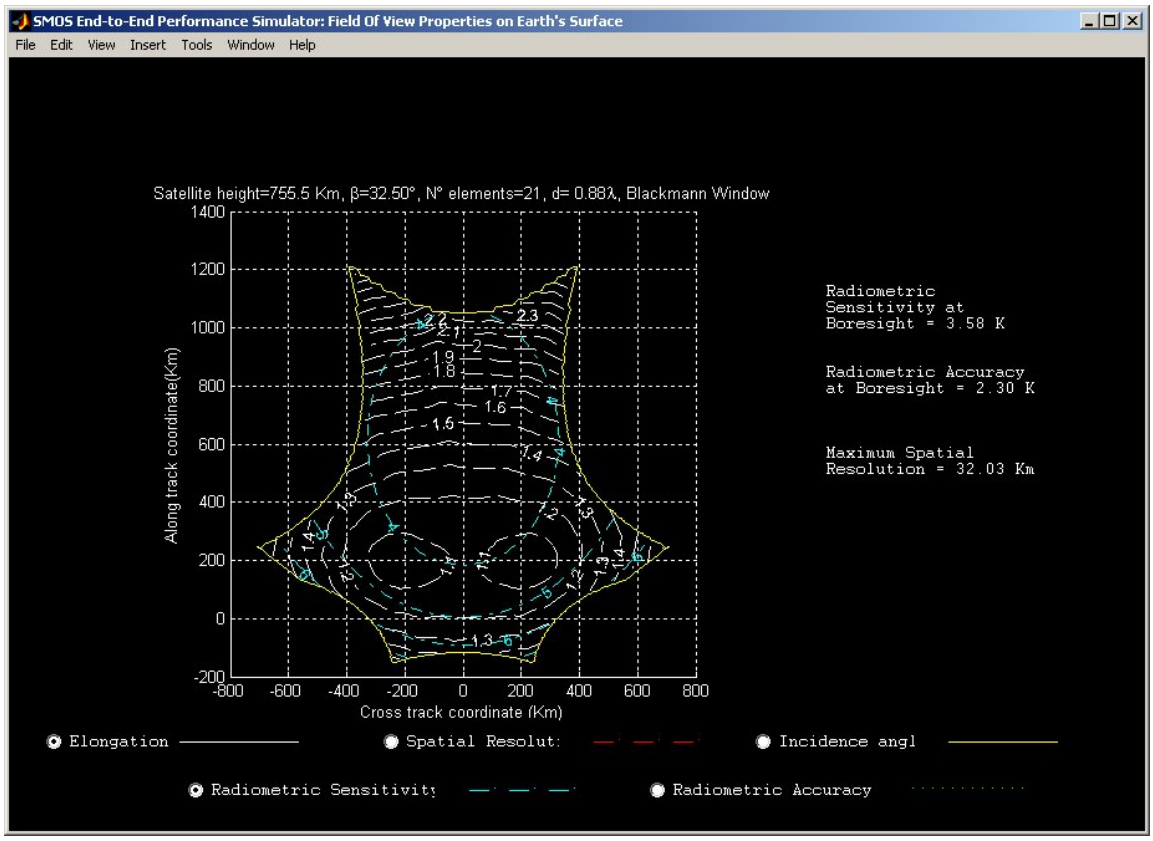


Figure VIII.15. The figure after disabling three of the parameters plotted, through the radio buttons displayed below the plot.

The third window is shown in Figure VIII.16 is displayed, containing the value for the computed synthesized half-power beamwidth. The user may press the ***OK*** push button to close the window.

Figure VIII.16. The third results figure of the FOV Properties tool shows the computed value for the synthesized half-power beamwidth.

As it has been said above, occasionally the figures presented can be only two. This happens when the platform is pointing towards the sky and therefore not having the Earth's shape in its scope. Then, the second figure containing the FOV properties over the Earth's surface is obviously not presented. In the figure containing the properties in director cosines domain, the FOV in yellow color is limited then by the unit circles of the alias images. Also the hexagon in white color is plotted, although in this case is not limiting the FOV. Figure VIII.17 displays the resulting figure.

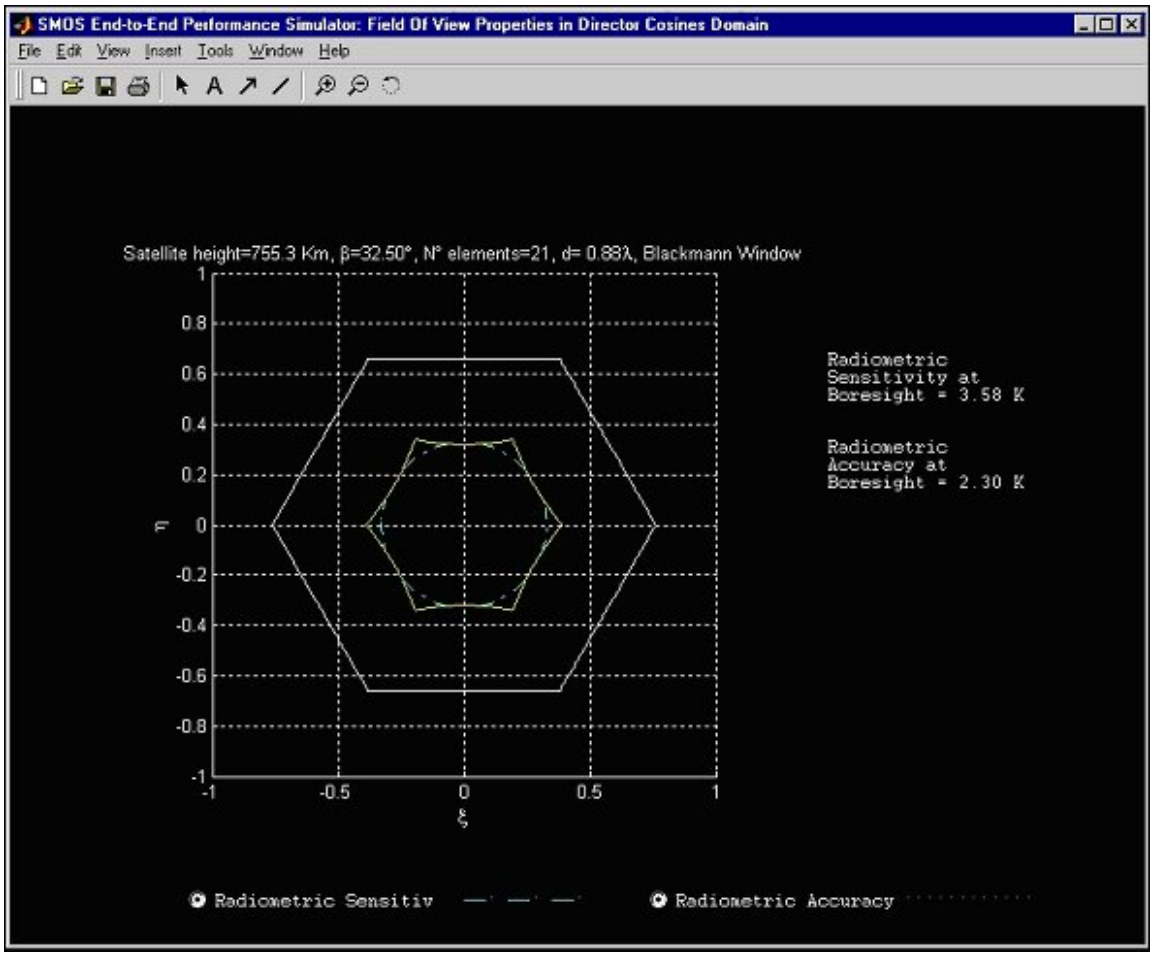


Figure VIII.17. The first results figure of the FOV Properties tool changes when pointing towards the sky. The FOV is limited by the unit circles of the alias images. The second figure with the FOV Properties over the Earths surface is not presented, as Earth does not appear in the radiometer's scope.

VIII.5. Reliability

This tool is intended to allow evaluating the total reliability of the system, from the values given by the user to the reliabilities of the system's single devices.

VIII.5.1. Input Parameters

The user may press the ***Reliability*** push button in the upper part of the SEPS *FOV Calculator* interface in order to display the input parameters interface for the *Reliability* tool. The interface is seen if Figure VIII.18, and consists of two edit boxes where the user may modify the values if needed. The inputs to compute the total reliability of the system are the reliability of a single LICEF or receiver and also the reliability of a single Noise Injection Radiometer (NIR). Their values must be within the range of 0 and 1, and it is considered identical for all the devices of the same type.

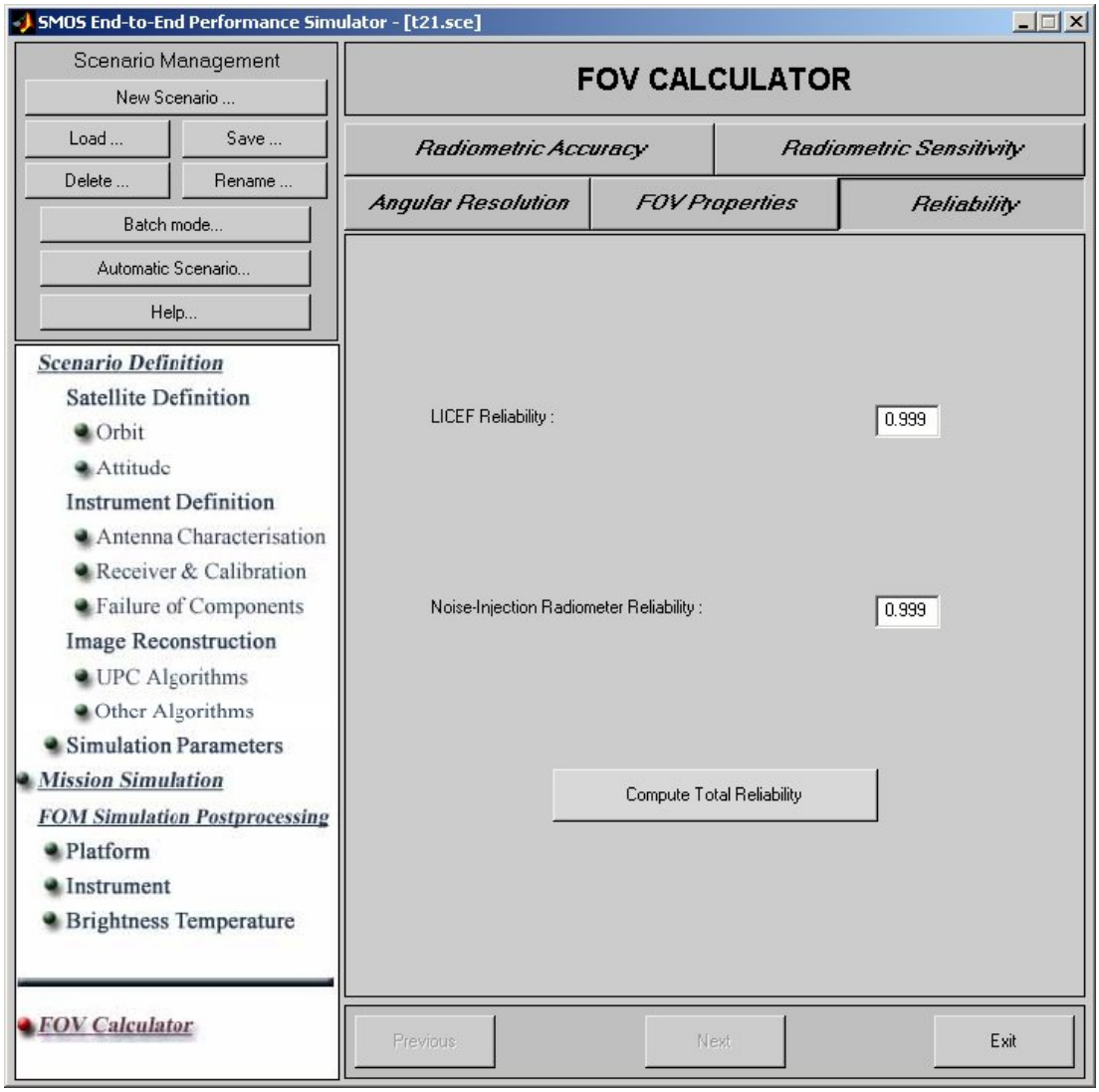


Figure VIII.18. Reliability *tool interface for updating the input parameters and launch the computation of the system's total reliability.*

VIII.5.2. Outputs

Clicking the ***Compute Total Reliability*** push button, the input parameters are used to obtain the system's reliability value, which is shown in a small figure opened for that purpose. This small window is shown in Figure VIII.19. The figure can be finished by pressing the ***Close*** push button.

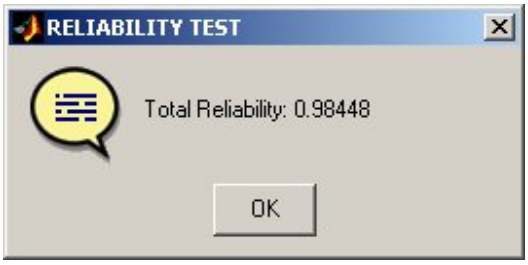


Figure VIII.19. Reliability *tool output showing the system's total reliability.*

CHAPTER IX. APPLICATIONS

This chapter describes the use of the resulting simulated data when not working with SEPS. Data is saved at the end of a simulation in text ASCII format and binary .mat format –if the reconstruction algorithm is used, selecting the corresponding option in the *Mission Simulation* interface-.

Also, science and ancillary packet files are saved to disk for later use –if the packets option is requested, selecting the corresponding option in the *Mission Simulation* interface-.

IX.1. Output ASCII and binary data files

The description of the resulting output text files of a simulation executing the image reconstruction algorithm option follows. These files are created in order the user can manage the data reading it in ASCII format. Once the reconstruction algorithm has been executed –if the suitable option has been selected in the *Mission Simulation* interface-, a routine to save all simulated data is executed. This routine creates two types of files, which can be found in the *SEPSroot / Data / Output / Bin / ScenarioName /* folder, being *SEPSroot* the SEPS root installation folder and *ScenarioName* the name given by the user to the scenario before launching the simulation. The two types of files are:

1. The first type of files, which are named as *ScenarioName_XXXX* (where *ScenarioName* is the name given by the user to the simulated scenario, and *XXXX* is a 4 digit number from 0001 to the maximum number of snapshots, that identifies the number of snapshot in the simulation), save the simulated data for the points in the director cosines domain regular mesh. The data is saved in text ASCII format –the data is already stored in MATLAB binary files, named as *data_numeric_XXXX.mat*-. The ASCII files have the appearance of the following figure:

t01_0001.txt - WordPad

Archivo Edición Ver Insertar Formato Ayuda

```
PIXEL                    =     N° OF PIXEL
LAT                      =     LATITUDE
LON                      =     LONGITUDE
Xi                       =     DIRECTOR COSINES XI COORDINATE
Eta                      =     DIRECTOR COSINES ETA COORDINATE
INC                      =     INCIDENCE ANGLE [°]
FROT                     =     FARADAY ROTATION INDUCED BY THE IONOSPHERE [°]
Geom.Rot.                =     GEOMETRIC ROTATION BETWEEN EARTH AND ANTENNA REFERENCE FRAMES
Azimuth                  =     AZIMUTH ANGLE OF THE PIXEL WITH RESPECT TO THE ALONG-TRACK DIRECTION
Attenuation              =     ATMOSPHERIC & IONOSPHERIC ATTENUATION IN dB
Spatial Res.             =     SPATIAL RESOLUTION OF THE PIXEL IN KM
TY_REC_ANT               =     RETRIEVED BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, Y POLARIZATION [K]
TX_REC_ANT               =     RETRIEVED BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, X POLARIZATION [K]
TYX_REC_ANT              =     RETRIEVED BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, YX POLARIZATION [K]
TY_REC_IDEAL_ANT         =     RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), ANTENNA REFERENCE FRAME, Y POLAF
TX_REC_IDEAL_ANT         =     RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), ANTENNA REFERENCE FRAME, X POLAF
TYX_REC_IDEAL_ANT        =     RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), ANTENNA REFERENCE FRAME, YX POLA
TY_ERROR_IDEAL-REC_ANT   =     ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, AN
TX_ERROR_IDEAL-REC_ANT   =     ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, AN
TYX_ERROR_IDEAL-REC_ANT  =     ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, AN
I_REC_ANT                =     FIRST STOKES PARAMETER I, COMPUTED USING THE RETRIEVED BRIGHTNESS TEMPERATURES,ANTENNA REFEREN
```

PIXEL	LAT	LON	Xi	Eta	INC	FROT	Geom.Rot.	Azimuth	Attenuation	Spatial
+1	+039.98	-009.91	+0.000	+0.000	+37.209	+0.832	+179.930	-18.188	+0.0004	+42.
+2	+040.05	-009.84	+0.009	+0.005	+37.563	+0.837	+179.131	-16.965	+0.0004	+43.
+3	+040.12	-009.76	+0.018	+0.010	+37.926	+0.841	+178.347	-15.751	+0.0004	+43.
+4	+040.20	-009.68	+0.027	+0.015	+38.298	+0.845	+177.578	-14.548	+0.0004	+43.
+5	+040.27	-009.60	+0.036	+0.021	+38.678	+0.849	+176.826	-13.356	+0.0004	+44.
+6	+040.34	-009.52	+0.045	+0.026	+39.067	+0.853	+176.088	-12.175	+0.0004	+44.
+7	+040.42	-009.44	+0.054	+0.031	+39.463	+0.857	+175.365	-11.005	+0.0004	+44.
+8	+040.49	-009.36	+0.063	+0.036	+39.868	+0.862	+174.657	-9.846	+0.0004	+45.
+9	+040.57	-009.27	+0.071	+0.041	+40.280	+0.866	+173.963	-8.699	+0.0004	+45.
+10	+040.65	-009.19	+0.080	+0.046	+40.701	+0.870	+173.284	-7.563	+0.0004	+45.
+11	+040.73	-009.10	+0.089	+0.052	+41.129	+0.874	+172.619	-6.439	+0.0004	+46.
+12	+040.80	-009.01	+0.098	+0.057	+41.565	+0.878	+171.967	-5.326	+0.0004	+46.
+13	+040.88	-008.93	+0.107	+0.062	+42.009	+0.883	+171.330	-4.225	+0.0004	+47.
+14	+040.97	-008.84	+0.116	+0.067	+42.460	+0.888	+170.705	-3.136	+0.0004	+47.
+15	+041.05	-008.75	+0.125	+0.072	+42.919	+0.893	+170.094	-2.058	+0.0004	+48.
+16	+041.13	-008.65	+0.134	+0.077	+43.385	+0.898	+169.495	-0.991	+0.0004	+48.
+17	+041.22	-008.56	+0.143	+0.082	+43.859	+0.903	+168.909	+0.064	+0.0004	+49.
+18	+041.30	-008.46	+0.152	+0.088	+44.341	+0.908	+168.335	+1.109	+0.0004	+50.
+19	+041.39	-008.37	+0.161	+0.093	+44.830	+0.913	+167.774	+2.142	+0.0004	+50.

Figure IX.1. ASCII file showing the output parameters in a regular mesh in the director cosines domain, as they are actually simulated. Open this file with WordPad or similar text editor for proper view. Text editors such as Matlab M-file editor or Notepad cannot display a large number of columns.

The introductory text explains which the data types in each tabulated column are. Each row contains all the data types for a particular point given by (Xi, Eta) –the director cosines domain co-ordinates-. The data types for each column, from left to right, are the following:

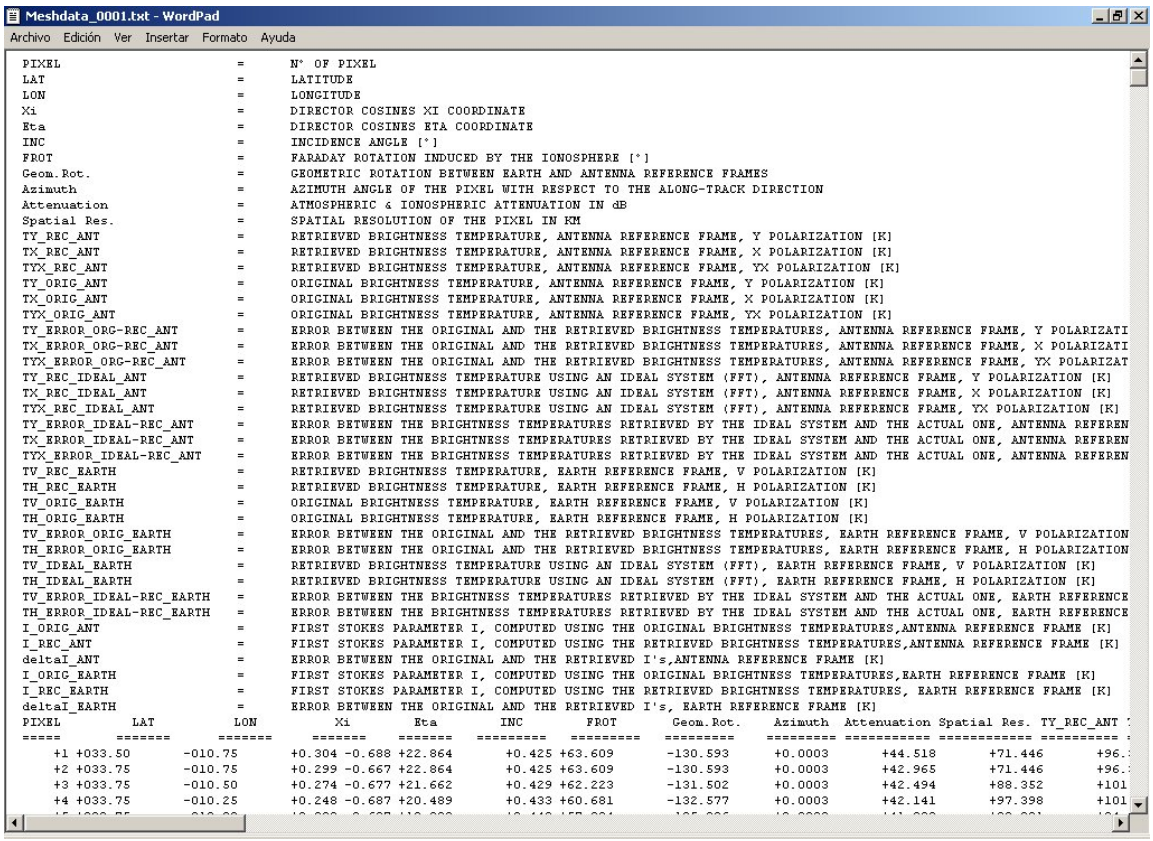
- *PIXEL*: the points of the image are numbered so as to identify them, but this numbering has no physical meaning.
- *LAT*: geographic latitude, in sexagesimal degrees.
- *LON*: geographic longitude, in sexagesimal degrees.
- *Xi*: ξ co-ordinate of the point in the director cosines domain.
- *Eta*: η co-ordinate of the point in the director cosines domain.
- *INC*: incidence angle, in sexagesimal degrees.

- *Far.Rot*: Faraday rotation angle value for the current pixel, in sexagesimal degrees.
- *Geo.Rot*: geometric polarization mixing, in sexagesimal degrees.
- *AZIM*: azimuth angle, in sexagesimal degrees.
- *ATT*: attenuation induced by the ionosphere, in dB.
- *SPRES*: spatial resolution, computed as the product of the major and minor axes of the ellipsoidal spot of the pixel on Earth's surface, in km.
- *TY(rec)*: brightness temperature using the selected image reconstruction algorithm, at the antenna reference frame, Y-polarization, in Kelvin.
- *TX(rec)*: brightness temperature using the selected image reconstruction algorithm, at the antenna reference frame, X-polarization, in Kelvin.
- *TXY(rec)*: brightness temperature using the selected image reconstruction algorithm, at the antenna reference frame, full polarimetric, in Kelvin.
- *TY(ant)*: ideal noise-free FFT-recovered brightness temperature at antenna reference frame, Y-polarization, in Kelvin.
- *TX(ant)*: ideal noise-free FFT-recovered brightness temperature at antenna reference frame, X-polarization, in Kelvin.
- *TYX(ant)*: ideal noise-free FFT-recovered brightness temperature at antenna reference frame, Cross-polarization, in Kelvin.
- *TY(ant)-TY(rec)*: brightness temperature error in Y-polarization, computed as the difference between *TYant* and *TY(rec)*, in Kelvin.
- *TX(ant)-TX(rec)*: brightness temperature error in X-polarization, computed as the difference between *TXant* and *TX(rec)*, in Kelvin.
- *TYX(ant)-TYX(rec)*: brightness temperature error in Cross-polarization, computed as the difference between *TYXant* and *TXY(rec)*, in Kelvin.
- *STOKES I*: Value of Stokes I parameter, computed as the sum of *TX(rec)* and *TY(rec)*, in Kelvin.

2. The second type of files, which are named as *Meshdata_snapshot_XXXX* (where *XXXX* is a 4 digit number from 0001 to the maximum number of snapshots, that identifies the number of snapshot in the simulation), creates a regular mesh of geographic latitude and longitude points. The size of this grid can be selected by

means of the Cell's resolution box in Mission Simulation menu. The simulated data is interpolated to fit this new mesh.

This pre-processing is done to provide the cartographic projections module with maps that can be immediately projected without any previous computation. The data is saved in MATLAB binary .mat format –to make it available for loading in the MATLAB environment- and in text ASCII format. Also a pair of files *Meshdata_total.mat /.txt* is created, in order to project a global map with all the snapshots of the simulation super-imposed in the cartography tool. The text files have the appearance of the Figure IX.2:



```
Meshdata_0001.txt - WordPad
Archivo  Edición  Ver  Insertar  Formato  Ayuda

PIXEL                          =      N° OF PIXEL
LAT                            =      LATITUDE
LON                            =      LONGITUDE
Xi                             =      DIRECTOR COSINES XI COORDINATE
Eta                            =      DIRECTOR COSINES ETA COORDINATE
INC                            =      INCIDENCE ANGLE [°]
FROT                           =      FARADAY ROTATION INDUCED BY THE IONOSPHERE [°]
Geom.Rot.                      =      GEOMETRIC ROTATION BETWEEN EARTH AND ANTENNA REFERENCE FRAMES
Azimuth                        =      AZIMUTH ANGLE OF THE PIXEL WITH RESPECT TO THE ALONG-TRACK DIRECTION
Attenuation                    =      ATMOSPHERIC & IONOSPHERIC ATTENUATION IN dB
Spatial Res.                   =      SPATIAL RESOLUTION OF THE PIXEL IN KM
TY_REC_ANT                     =      RETRIEVED BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, Y POLARIZATION [K]
TX_REC_ANT                     =      RETRIEVED BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, X POLARIZATION [K]
TYX_REC_ANT                    =      RETRIEVED BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, YX POLARIZATION [K]
TY_ORIG_ANT                    =      ORIGINAL BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, Y POLARIZATION [K]
TX_ORIG_ANT                    =      ORIGINAL BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, X POLARIZATION [K]
TYX_ORIG_ANT                   =      ORIGINAL BRIGHTNESS TEMPERATURE, ANTENNA REFERENCE FRAME, YX POLARIZATION [K]
TY_ERROR_ORG-REC_ANT           =      ERROR BETWEEN THE ORIGINAL AND THE RETRIEVED BRIGHTNESS TEMPERATURES, ANTENNA REFERENCE FRAME, Y POLARIZATI
TX_ERROR_ORG-REC_ANT           =      ERROR BETWEEN THE ORIGINAL AND THE RETRIEVED BRIGHTNESS TEMPERATURES, ANTENNA REFERENCE FRAME, X POLARIZATI
TYX_ERROR_ORG-REC_ANT          =      ERROR BETWEEN THE ORIGINAL AND THE RETRIEVED BRIGHTNESS TEMPERATURES, ANTENNA REFERENCE FRAME, YX POLARIZAT
TY_REC_IDEAL_ANT               =      RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), ANTENNA REFERENCE FRAME, Y POLARIZATION [K]
TX_REC_IDEAL_ANT               =      RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), ANTENNA REFERENCE FRAME, X POLARIZATION [K]
TYX_REC_IDEAL_ANT              =      RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), ANTENNA REFERENCE FRAME, YX POLARIZATION [K]
TY_ERROR_IDEAL-REC_ANT         =      ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, ANTENNA REFEREN
TX_ERROR_IDEAL-REC_ANT         =      ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, ANTENNA REFEREN
TYX_ERROR_IDEAL-REC_ANT        =      ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, ANTENNA REFEREN
TV_REC_EARTH                   =      RETRIEVED BRIGHTNESS TEMPERATURE, EARTH REFERENCE FRAME, V POLARIZATION [K]
TH_REC_EARTH                   =      RETRIEVED BRIGHTNESS TEMPERATURE, EARTH REFERENCE FRAME, H POLARIZATION [K]
TV_ORIG_EARTH                  =      ORIGINAL BRIGHTNESS TEMPERATURE, EARTH REFERENCE FRAME, V POLARIZATION [K]
TH_ORIG_EARTH                  =      ORIGINAL BRIGHTNESS TEMPERATURE, EARTH REFERENCE FRAME, H POLARIZATION [K]
TV_ERROR_ORIG_EARTH            =      ERROR BETWEEN THE ORIGINAL AND THE RETRIEVED BRIGHTNESS TEMPERATURES, EARTH REFERENCE FRAME, V POLARIZATION
TH_ERROR_ORIG_EARTH            =      ERROR BETWEEN THE ORIGINAL AND THE RETRIEVED BRIGHTNESS TEMPERATURES, EARTH REFERENCE FRAME, H POLARIZATION
TV_IDEAL_EARTH                 =      RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), EARTH REFERENCE FRAME, V POLARIZATION [K]
TH_IDEAL_EARTH                 =      RETRIEVED BRIGHTNESS TEMPERATURE USING AN IDEAL SYSTEM (FFT), EARTH REFERENCE FRAME, H POLARIZATION [K]
TV_ERROR_IDEAL-REC_EARTH       =      ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, EARTH REFERENCE
TH_ERROR_IDEAL-REC_EARTH       =      ERROR BETWEEN THE BRIGHTNESS TEMPERATURES RETRIEVED BY THE IDEAL SYSTEM AND THE ACTUAL ONE, EARTH REFERENCE
I_ORIG_ANT                     =      FIRST STOKES PARAMETER I, COMPUTED USING THE ORIGINAL BRIGHTNESS TEMPERATURES,ANTENNA REFERENCE FRAME [K]
I_REC_ANT                      =      FIRST STOKES PARAMETER I, COMPUTED USING THE RETRIEVED BRIGHTNESS TEMPERATURES,ANTENNA REFERENCE FRAME [K]
deltaI_ANT                     =      ERROR BETWEEN THE ORIGINAL AND THE RETRIEVED I's,ANTENNA REFERENCE FRAME [K]
I_ORIG_EARTH                   =      FIRST STOKES PARAMETER I, COMPUTED USING THE ORIGINAL BRIGHTNESS TEMPERATURES,EARTH REFERENCE FRAME [K]
I_REC_EARTH                    =      FIRST STOKES PARAMETER I, COMPUTED USING THE RETRIEVED BRIGHTNESS TEMPERATURES, EARTH REFERENCE FRAME [K]
deltaI_EARTH                   =      ERROR BETWEEN THE ORIGINAL AND THE RETRIEVED I's, EARTH REFERENCE FRAME [K]
PIXEL        LAT          LON          Xi        Eta        INC        FROT       Geom.Rot.    Azimuth  Attenuation Spatial Res. TY_REC_ANT
=====      =======      =======      =======   =======   =========  =========   =========   ========= =========== ============ =========
    +1 +033.50      -010.75      +0.304 -0.688 +22.864      +0.425 +63.609      -130.593     +0.0003      +44.518      +71.446      +96.
    +2 +033.75      -010.75      +0.299 -0.667 +22.864      +0.425 +63.609      -130.593     +0.0003      +42.965      +71.446      +96.
    +3 +033.75      -010.50      +0.274 -0.677 +21.662      +0.429 +62.223      -131.502     +0.0003      +42.494      +88.352      +101
    +4 +033.75      -010.25      +0.248 -0.687 +20.489      +0.433 +60.681      -132.577     +0.0003      +42.141      +97.398      +101
```

Figure IX.2. ASCII file showing the output parameters in a regular mesh in geographic co-ordinates. This is done to pre-process data for the cartographic projections module.

The ASCII files show for each column the following variables:

- *PIXEL*: the points of the image are numbered so as to identify them, but this numbering has no physical meaning.
- *LAT*: geographic latitude, in sexagesimal degrees.
- *LON*: geographic longitude, in sexagesimal degrees.
- *Xi*: ξ co-ordinate of the point in the director cosines domain.
- *Eta*: η co-ordinate of the point in the director cosines domain.
- *INC*: incidence angle, in sexagesimal degrees.
- *Far.Rot*: Faraday rotation angle value for the current pixel, in sexagesimal degrees.
- *Geo.Rot*: geometric polarization mixing, in sexagesimal degrees.
- *AZIM*: azimuth angle, in sexagesimal degrees.
- *ATT*: attenuation induced by the ionosphere, in dB.
- *SPRES*: spatial resolution, computed as the product of the major and minor axes of the ellipsoidal spot of the pixel on Earth's surface, in km.
- *TY(rec)*: brightness temperature using the selected image reconstruction algorithm, at the antenna reference frame, Y-polarization, in Kelvin.
- *TX(rec)*: brightness temperature using the selected image reconstruction algorithm, at the antenna reference frame, X-polarization, in Kelvin.
- *TXY(rec)*: brightness temperature using the selected image reconstruction algorithm, at the antenna reference frame, full polarimetric, in Kelvin.
- *TY(orig)*: original brightness temperature arriving at the antenna reference frame, Y-polarization, in Kelvin.
- *TX(orig)*: original brightness temperature arriving at the antenna reference frame, X-polarization, in Kelvin.
- *TYX(orig)*: original brightness temperature arriving at the antenna reference frame, Cross-polarization, in Kelvin.
- *deltaTYrec-TYorg*: brightness temperature difference between the reconstructed temperature and the original temperature at the antenna reference frame, Y-polarization, in Kelvin.
- *deltaTXrec-TXorg*: brightness temperature difference between the reconstructed temperature and the original temperature at the antenna reference frame, X-polarization, in Kelvin.

- *deltaTYXrec-TYXorg*: brightness temperature difference between the reconstructed temperature and the original temperature at the antenna reference frame, Cross-polarization, in Kelvin.
- *TY(ideal)*: brightness temperature retrieved with an ideal system using an Inverse FFT as image reconstruction algorithm, at the antenna reference frame, Y-polarization, in Kelvin.
- *TX(ideal)*: brightness temperature retrieved with an ideal system using an Inverse FFT as image reconstruction algorithm, at the antenna reference frame, X-polarization, in Kelvin.
- *TXY(ideal)*: brightness temperature retrieved with an ideal system using an Inverse FFT as image reconstruction algorithm, at the antenna reference frame, full polarimetric, in Kelvin.
- *deltaTYrec-TYideal*: brightness temperature difference between the reconstructed temperature and the ideal temperature at the antenna reference frame, Y-polarization, in Kelvin.
- *deltaTXrec-TXideal*: brightness temperature difference between the reconstructed temperature and the ideal temperature at the antenna reference frame, X-polarization, in Kelvin.
- *deltaTYXrec-TYXideal*: brightness temperature difference between the reconstructed temperature and the ideal temperature at the antenna reference frame, Cross-polarization, in Kelvin.
- *TV(recEarth)*: brightness temperature using the selected image reconstruction algorithm, at the Earth reference frame, V-polarization, in Kelvin.
- *TH(recEarth)*: brightness temperature using the selected image reconstruction algorithm, at the Earth reference frame, H-polarization, in Kelvin.
- *TV(origEarth)*: original brightness temperature at the Earth reference frame, V-polarization, in Kelvin.
- *TH(origEarth)*: original brightness temperature at the Earth reference frame, H-polarization, in Kelvin.
- *deltaTVrecEarth-TVorigEarth*: brightness temperature error in V-polarization at the Earth reference frame, computed as the difference between *TVrecEarth* and *TVorigEarth*, in Kelvin.

- *deltaTHrecEarth-THorigEarth*: brightness temperature error in H-polarization at the Earth reference frame, computed as the difference between *THrecEarth* and *THorigEarth*, in Kelvin.
- *TV(idealEarth)*: brightness temperature retrieved with an ideal system using an Inverse FFT as image reconstruction algorithm, at the Earth reference frame, V-polarization, in Kelvin.
- *TH(idealEarth)*: brightness temperature retrieved with an ideal system using an Inverse FFT as image reconstruction algorithm, at the Earth reference frame, H-polarization, in Kelvin.
- *deltaTVidealEarth-TVrecEarth*: brightness temperature error in V-polarization at the Earth reference frame, computed as the difference between *TVidealEarth* and *TVrecEarth*, in Kelvin.
- *deltaTHidealEarth-THrecEarth*: brightness temperature error in H-polarization at the Earth reference frame, computed as the difference between *THidealEarth* and *THrecEarth*, in Kelvin.
- *I_orig_ant*: Stokes I parameter computed as the sum of the original brightness temperature in X and Y-polarizations, at the antenna reference frame, in Kelvin.
- *I_rec_ant*: Stokes I parameter computed as the sum of the retrieved brightness temperature in X and Y-polarizations, at the antenna reference frame, in Kelvin.
- *deltaIorig-Irec_ant*: error in the Stokes I parameter computed as the difference between the values for the original and the retrieved temperatures, at the antenna reference frame, in Kelvin.
- *I_orig_earth*: Stokes I parameter computed as the sum of the original brightness temperature in X and Y-polarizations, at the Earth reference frame, in Kelvin.
- *I_rec_earth*: Stokes I parameter computed as the sum of the retrieved brightness temperature in X and Y-polarizations, at the Earth reference frame, in Kelvin.
- *deltaIorig-Irec_Earth*: error in the Stokes I parameter computed as the difference between the values for the original and the retrieved temperatures, at the Earth reference frame, in Kelvin.

It can be clearly seen that geographic latitude and longitude points are now separated in regular intervals.

To load variables from the data files in binary format, the following MATLAB sentence can be used:

>> load ScenarioName_XXXX.mat variable1 variable 2;

while typing *load ScenarioName_XXXX.mat* directly load all the variables in the binary file. All variables in these binary .mat files are two-dimensional matrices of the reconstructed image size (set up in the *ξ-η resolution* edit box of the *Simulation Definition* interface). Their names are not the same as the ones listed above, but they are equivalent. The storage of binary and ASCII data is performed in function pixdata_uniform.m.

IX.2. Science and ancillary packets

If the packets option is selected –either choosing the *Packets Generation* or the *Image Reconstruction & Packets Generation* in the *Mission Simulation* interface-, the SEPS simulation stores two types of packet files following the format described in SO-TN-CASA-PLM-0279 and SO-TN-CASA-PLM-0549, for each switch position. These types of packet files are science data files and ancillary data files. A third type generated by the PLM and related with PLM event data, has not a purpose related to SEPS and therefore it is not modelled.

The packets can be found in the *SEPSroot / Data / Output / Bin / ScenarioName / Packets / SwitchPosition /* folders, where *SEPSroot* is the SEPS main folder, *ScenarioName* is the name given by the user to the scenario through the *Scenario Management* box, and *SwitchPosition* is the position of the receiver's switch for each type of measurement. *SwitchPosition* can be either *H*, *V*, *Ceven*, *Codd*, *U, VH or HV*. Therefore, 7 subfolders containing the computed packets exist, one for each switch position.

The science data, including all the computed correlations for a snapshot, is stored in the files named as *packets_XXXX_01*, where *XXXX* is the number of snapshot (starting in 0000 and ending in N-1, being N the maximum number of packets generated in the simulation):

There is also one ancillary data packet per science packet generated and switch position, containing ancillary data required for the image reconstruction, as well as PLM housekeeping data. This packet file is named as packets_XXXX_00.

The forth version of SEPS is an event-driver simulator and not a time-driver as it occurred in the previous versions. It implies that for a single event a set of packets for different epoch can be generated. In fact, for each type of event the following packets are generated:

- Calibration, each event is based in the computation of 4 consecutives points (separate 1.2 seconds). For each of these points the measurements corresponding to the different configuration adopted by the receiver are measured and four ancillary and science packets are generated. Namely, let be t_E the epoch for an event of calibration, then 4 packets are generated for the epochs [t_E, t_E+1.2, t_E+2.4, t_E+3.6], and they are numbered consecutively from N to N+3.
- Snapshot, this event computes the H- and V-polarization measurements (corresponding to the positions H and V of the receiver's switch). These measurements are collected in a real situation at consecutives epochs separated 1.2 seconds, but SEPS generate both measurement at the same time, generating only one packet.
- FTT Initialization and G-Matrix computation events. These events are executed in a single epoch, therefore the measurements computed during this epoch are stored directly to the ancillary and science packets.

In fact for each event (independently of its type, calibration and/or snapshot) the data corresponding to the seven positions of the receiver's switch are always computed (without considering the real switch position of the receiver U, V, H, HV, VH, Ceven or Codd). Therefore for the different types of receiver's switch position, all the data are

always available and for each epoch of each event (4 for calibration, 1 for snapshot and 1 for FTT initialization and G-matrix computation) the corresponding packets are generated.

***Note1**: The AOCS, PVT and OBET are computed for each epoch ([t_E, t_E+1.2, t_E+2.4, t_E+3.6] for calibration), however the science data corresponding to the different positions of the receiver's switch are computed only at the event epoch (the science packets are filled with the data computed in the Observable Generator module at epoch t_E –event time-).*

***Note2**: The tool allows the user to define events starting at any time, thus allowing the creation of data gaps, time overlap or even consecutive events that are spaced at less than 1.2 seconds. It has be done to allow full flexibility to the user and save a lot of amount of computational time, but it is left at the user's best judgement to define scenarios which are representative of the actual PLM behaviour: events only occur at epoch, which are spaced at 1.2 seconds intervals.*

***Note3**: From a packet generation point of view, the different events involve different receiver's switch position, and therefore are incompatibles. If the user want use SEPS to generate packets then shall define the events in the timeline for that they are not coincident at time (taking account that a calibration event takes 4*1.2=4.8 seconds, a snapshot event takes 1.2 seconds and the FTT initialization and G-matrix computation only takes 1.2 seconds).*

The packets format can be changed by editing the *Packets.txt* file in the *SEPSroot / Data / Defaults /* folder. This file contains information on how the packets are internally organized. **Only advanced users should dare editing this file**.

IX.3. Changing system's definition database

The user may want to evaluate the impact on system's performances of changing certain characteristics of the instrument or the orbit, using input files whenever possible. Although all these operations are described in other sections of this document, a summary of the most common changes may be useful:

- *Change of orbit and attitude*: the orbital characteristics of the platform, as well as its attitude, can be changed in the Satellite Definition section. The possibility of changing the orbital motion and platform's attitude files is available. The formats needed for the user files –if this option is preferred instead of using the interfaces- are reviewed in sections V.1.1 and V.1.3, respectively. Samples of these files –*orbitalmotion.dat* and *attitudeprofile.dat, as well as the default OSF SM__MPL_ORBREF_20070223T060002_99999999T999999_0000.EEF-* can be found in the *SEPSroot / Data / Input / Defaults /* folder.
- The *change of antenna positions* can be performed by supplying a new systematic errors file. This file must contain the following format: after an introductory text, four columns containing the number of LICEF, and the position errors in metres, in-plane X, in-plane Y, and off-plane Z. See default file *IOposerr_oct04.dat and OGCposerr_oct04.dat* in *SEPSroot / Data / Defaults /* folder to have an example.
- The *change of temperatures*' file can be performed in the *Instrument Definition* section, *Antenna Characterisation* subsection, *Pattern and Thermal* interface. The format of this files, that can be supplied either in Excel .xls or in ASCII .dat file types, must follow the format described in section V.2.1 of this document. A sample of these files can be seen in the *SEPSroot / Data / Input / Defaults /* folder –*SampleTemps.xls* and *SampleTemps.dat*-.
- The *change of receiver characteristics* can be also performed by advanced users. The receiver characteristics' database is in the *SEPSroot / Data / Input / DefaultInputFiles / Receiver /* folder. A review of the necessary formats for the files is provided in the *Appendix A. Modifying Receivers' Parameters* of the *Advanced Topics* part of this document.
- The *change of antenna characteristics* can be performed substituting the files in the SEPS database, in the *SEPSroot / Data / Input / DefaultInputFiles / AntennaPattern /* folder. This operation should only be performed by advanced users. The antenna patterns' new database can be supplied either in text files or in binary files. See the *Appendix A. Modifying Receivers' Parameters*, section A.1, of the *Advanced Topics* part of this document.

- The *change of the Noise Injection Radiometer characteristics* can be performed by substituting the *NIR.sbs* file in the *SEPSroot / Data / Input / DefaultInputFiles / Receiver /* folder. Generally, all the receiver's subsystems characteristics can be changed following the formats and instructions reviewed in the *Appendix A. Modifying Receivers' Parameters*, of the *Advanced Topics* part of this document.

APPENDICES:

ADVANCED TOPICS

APPENDIX A. MODIFYING RECEIVERS' PARAMETERS

In addition to the SEPS v1.0 option for computing the receivers' frequency response from its blocks characteristics –which is reviewed in the following sections-, also the SEPS v2.1 provided the alternative of loading already computed files from the database containing the receivers' frequency responses. Since SEPS v3.0 these receiver's frequency responses are the measured ones for a set of 10 receivers, repeated as many times as needed to obtain the receiver's frequency responses of 90 receivers. The user may choose the use of one of them in the *Receiver & Calibration* interface of the *Instrument Definition* section in the *Scenario Definition* box.

The following sections are aimed to explain the alternative already present in SEPS v1.0, that is, the receivers' frequency responses are computed by cascading the parameters of their individual subsystems taking into account that each one is slightly different from each other.

For each subsystem, the nominal characterizing parameters and their deviations are read from text files with the extension **'.sbs'**. Comments can be included in the files using the character **"%"** as standard MATLAB code and the parameters can be edited to customize the computations.

All files except those of the two filters (RF and IF) have the information in tabular form with the relevant parameters as a function of frequency, as described below:

Frequency	**Parameter #1**	**Parameter #2**		**Parameter #m**
f_1 f_2 … f_n	Value at f_1 Value at f_2 … Value at f_n	Value at f_1 Value at f_2 … Value at f_n	… … … …	Value at f_1 Value at f_2 … Value at f_n
0	rms deviation	rms deviation	…	rms deviation
0	Temp. coefficient	Temp. coefficient	…	Temp. coefficient

Note ➔ Shortly after the release of SEPS v.4.0 there will be an update of the DB, in which Flight LICEF data will be included

The frequency is always expressed in MHz, being this referred to RF for all subsystems from antenna to mixer included, and to IF for the remaining. The number of frequency points and their value is defined in the AD/DD document (SO-TN-GMV-PLM-0003) for each subsystem.

The meaning and number of the parameters in the above table depends on the specific subsystem, as detailed for each one in the following sections. In any case, the parameter values are those corresponding to the nominal physical temperature, (25°C).

The last two rows contain the *rms deviation* and *temperature coefficient* information and they are identified by a value of 0 (zero) in the frequency column. Both deviation and temperature coefficient are assumed frequency independent and defined as follows.

a) For amplitude parameters

The rms deviation of a given parameter of value p has units of percent (%) and is defined in a relative basis:

$$\text{rms deviation}=\frac{\sigma_p}{\bar{p}}*100 \qquad \text{(B.1)}$$

where $\sigma_p = \sqrt{\overline{(p-\bar{p})^2}}$ is the standard deviation of the parameter value and the bar stands for "average value" along all receivers. The value of the parameter p in the above formula is in all cases expressed in *linear scale* at the nominal temperature (25°C) even in the cases in which it is expressed in logarithmic scale dB in the .sbs file.

The temperature coefficient of a parameter of value p has units of percent per degree centigrade (%/°C) and is defined as:

$$\text{Temperature coefficient}=\frac{p(T)-p(T_0)}{T-T_0}\frac{1}{p(T_0)}*100 \qquad \text{(B.2)}$$

where p(T) is the parameter value at temperature T and T_0 is the nominal temperature (25°C).

b) For phase parameters

In the case of phase, the deviation and temperature coefficient are defined in an absolute basis instead of relative. Then the rms deviation is defined simply as:

$$\text{rms deviation}=\sigma_p \qquad \text{(B.3)}$$

being σ_p the same as above. In this case the deviation has the units of deg.

The temperature coefficient for phase is defined as:

$$\text{Temperature coefficient}=\frac{p(T)-p(T_0)}{T-T_0} \qquad \text{(B.4)}$$

with the same meaning as above. The units in this case are deg/°C.

From the information above and the receivers' physical temperatures, the simulator computes the parameters' values (p) for each subsystem in the different receivers. In the following sections the file names for all subsystems and parameters included in them are given.

A.1. Antenna

The antenna parameters as a receiver subsystem are given in the two following text files:

```
AntennaH.sbs
AntennaV.sbs
```

being the parameters included in them the following:

Parameter #1: Magnitude of Reflection coefficient
Parameter #2: Phase of Reflection coefficient (deg)
Parameter #3 Antenna Ohmic efficiency (dB)=-10*log[$P_{radiated}/P_{input}$]

Antenna pattern parameters can be input from the GUI interface (used to evaluate a analytical function) or from a set of files containing measured or simulated data in MATLAB binary format:

- For the antenna patterns used "on ground" to compute the auxiliary visibilities and in the image reconstruction algorithm:

  `og_casa_patch_cop_x.mat`
  `og_casa_patch_cop_y.mat`
  `og_casa_patch_cross_x.mat`
  `og_casa_patch_cross_y.mat`

  And, for the backlobe antenna pattern:

  `og_casa_patch_back_cop_x.mat`
  `og_casa_patch_back_cop_y.mat`
  `og_casa_patch_back_cross_x.mat`
  `og_casa_patch_back_cross_y.mat`

- For the antenna patterns used "in orbit" to compute the instruments' observables (visibilities measured by the instrument):

  `io_patch_cop_x.mat`
  `io_patch_cop_y.mat`
  `io_patch_cross_x.mat`
  `io_patch_cross_y.mat`

  And, for the backlobe antenna pattern:

  `io_patch_back_cop_x.mat`

io_patch_back_cop_y.mat
io_patch_back_cross_x.mat
io_patch_back_cross_y.mat

A.2. Subsystems defined by S-parameters

The following is the list of files that contain S-parameter data:

switchH.sbs
switchV.sbs
switchC.sbs
switchU.sbs
isolator.sbs
lna.sbs
rfamp.sbs
coupler.sbs
ifattenuator.sbs

All the files except those of the switch correspond to 2-port subsystems (for the coupler it is only considered the main path), and the order of columns is the standard one used in circuit analysis software:

Parameters #1 and #2: Magnitude and phase (deg) of S_{11}
Parameters #3 and #4: Magnitude and phase (deg) of S_{21}
Parameters #5 and #6: Magnitude and phase (deg) of S_{12}
Parameters #7 and #8: Magnitude and phase (deg) of S_{22}

In the case of the switch, there are four files corresponding to the four positions of the switch. The ports are numbered as:

Port 1: H input
Port 2: V input
Port 3: C input

Port 4: U input

Port 5: Output.

And the following parameters are included in each of the files:

a) file `switchH.sbs` (through connection between ports 1 and 5)

Parameters #1 and #2: Magnitude and phase (deg) of S_{11}

Parameters #3 to #8: Magnitude and phase of S_{51} to S_{55}

b) file `switchV.sbs` (through connection between ports 2 and 5)

Parameters #1 and #2: Magnitude and phase (deg) of S_{22}

Parameters #3 to #8: Magnitude and phase of S_{51} to S_{55}

c) file `switchC.sbs` (through connection between ports 3 and 5)

Parameters #1 and #2: Magnitude and phase (deg) of S_{33}

Parameters #3 to #8: Magnitude and phase of S_{51} to S_{55}

d) file `switchU.sbs` (through connection between ports 4 and 5)

Parameters #1 and #2: Magnitude and phase (deg) of S_{44}

Parameters #3 to #8: Magnitude and phase of S_{51} to S_{55}

A.3. Subsystems defined by voltage gain (and input impedance)

The following files refer to subsystems defined by voltage gain and (one of them) input impedance:

`slopecorrector.sbs`

`ifamp1.sbs`

`ifamp2.sbs`

`ifamp3.sbs`

The parameters included in the files are:

Parameter #1: Voltage gain amplitude
Parameter #2: Voltage gain phase
Parameter #3: Input impedance (Ω). (only for the slope corrector)

A.4. Subsystems with specific noise descriptions (amplifiers)

The low noise amplifier (LNA) and the RF amplifier have specific noise descriptions in the following files:

```
lnanoise.sbs
rfampnoise.sbs
```

In this case only one parameter is included in the .sbs file:

Parameter #1: Noise Figure (dB)

A.5. Frequency converter (I/Q mixer)

This is a special component, since its description has to take into account the frequency conversion. The corresponding file is

```
mixer.sbs
```

and includes the following parameters:

Parameter #1: Conversion Gain I branch (dB)
Parameter #2: Conversion Gain Q branch (dB)
Parameter #3: Phase difference at IF: phase(I)-phase(Q) (deg)
Parameter #4: locking phase error of the synthesizer circuit (deg)
Parameter #5: Input reflection coefficient (Magnitude)
Parameter #6: Input reflection coefficient (Angle deg)
Parameter #7: I-branch Output reflection coefficient (Magnitude)
Parameter #8: I-branch Output reflection coefficient (Angle deg)

Parameter #9: Q-branch Output reflection coefficient (Magnitude)

Parameter #10: Q-branch Output reflection coefficient (Angle deg)

The last two parameters (output reflection coefficients) are function of the IF frequency (=RFfreq-LOfreq), even though the first column of the data is the RF frequency. The parameter #4 is referred to the phase error of the local oscillator and is independent of frequency. For the sake of format compatibility the same value is repeated for all frequencies.

A.6. RF filter

The RF filter is not described as a set frequency dependent parameters because its S-parameters are computed using circuit analysis techniques assuming a Chevichef filter made of transmission line resonators and inverters (SO-TN-GMV-PLM-0003). All the relevant information is included in the file

`rffilter.sbs`

which includes the following parameters in a single column:

Filter order.

Nominal ripple in dB

Lower frequency of band pass (MHz)

Higher frequency of band pass (MHz)

mean value of resonators Q

rms deviation of inverter constants (%)

rms deviation of resonators resonant frequency (%).

rms deviation of Q factor of resonators (%).

rms deviation of characteristic impedance of resonators (%)

Temperature coefficient of inverter constants (%/°C)

Temperature coefficient of resonators resonant frequency (%/°C)

Temperature coefficient of Q factor (%/°C)

Temperature coefficient of characteristic impedance of resonators (%/°C)

The values for lower and higher frequencies of the band pass *are not the nominal ones* but those computed to produce the best fit between simulated and measured filter frequency responses. They should not be modified unless an alternate filter design is envisaged.

The rms deviations and the temperature coefficient appearing in the file have the definitions given at the beginning of this section.

A.7. IF filter

The IF filter has a similar description as the RF filter, but applied to the corresponding design, being this a lumped element Chevichef band pass filter of order 3. The parameters are given in file

`Iffilter.sbs`

including the following data in a single column:

Nominal Q factor of resonators
rms deviation of inductors (%).
rms deviation of capacitors (%).
rms deviation of Q factor in (%).
Temperature coefficient of inductors (%/°C)
Temperature coefficient of capacitors (%/°C)
Temperature coefficient of Q factor (%/°C)

A.8. NIR

The NIR model has been updated significantly from previous versions. The model implemented in SEPS v.4.0 has been obtained from the latest data provided by HUT, as described in SO-TN-CASA-PLM-0854 v1.1, and it already has updated .sbs file.

The Noise Injection Radiometer (NIR) parameters must be loaded from the files:

NIR_A.sbs

NIR_R.sbs

For a description of these files, please see *SO-TN-CASA-PLM-0854 v1.1*

The function NIR_positions.m is used to localize the Noise Injection Radiometer in the hub. In the present configuration, the NIRs are the devices nearest to the center of the array (3, 3+NEL+NRED and 3+2(NEL+ NRED), where NEL is the number of antennas per arm and NRED is the number of redundant antennas per arm).

A.9. PMS

The Power Measurement System (PMS) parameters must be loaded from the file

PMS.sbs

creating a variable which consists of:

- first row: additional attenuation (= 6),
- second row: exponent (power of ten) of the a0 coefficients (non-linearity term),
- rows 3 to 74: mantissa of the a0 coefficients,
- rows 75 to 146: values of G0 parameter (mV/K): mean = 1.21 mV/K, standard deviation 0.03 mV/K
- row 147: exponent (power of ten) of the G0 parameters sensitivity to physical temperature,
- rows 148 to 219: mantissa of the G0 parameters sensitivity to physical temperature,
- rows 220 to 291: offset voltages: mean = 1.80 mV, standard deviation = 0.02 mV
- rows 292 to 363: sensitivity of offset voltages with respect to temperature.
- row 364: integration time of PMS (different from integration time for correlations) = 0.185 s.

Note: all these parameters can be updated by the user, or will be updated in future SEPS databases to include the upcoming measurements as they become available.

APPENDIX B. MODIFYING THE BRIGHTNESS TEMPERATURE GENERATOR

New brightness temperature models or new parameters can be implemented by modifying the function bright.m, where all the models are implemented and all the geophysical parameters data bases are loaded. In the present version of SEPS, the following geophysical parameters are loaded (see AD/DD document SO-TN-GMV-PLM-0003 for more details):

Table 1. Brightness temperature input parameters database.

Parameter	Units	File name
Rain rate	mm/h	bry88m12.prc
Atmospheric liquid water	mm	y88m12.cwp
Land and snow albedo	-	albedo.mat
Snow depth	inches	etac12.mat
Soil roughness	m	roughnss.mat
Sea surface salinity	psu	salero.mat
Vegetation albedo	-	sfy88m12.mat
Soil moisture	-	ssme812.mat
Surface temperature	K	ste81218.mat
Zonal wind speed	m/s	crwind.mat
Meridional wind speed	m/s	upwind.mat
Vegetation height	m	vege.mat
Sea ice cover fraction	%	y88m12i.mat
Sea surface temperature	K	y88m12.mat

Files with extension .mat are MATLAB binary files. Files with other extensions are flat text files. In SEPS v4.0 all geophysical parameters are given by a fix 180x360 elements matrix, except for rain rate, for which a bilinear interpolation to an 180x360 grid is performed in the same file. By replacing these files with others, or adding new ones, a new database can be created. Files can be MATLAB binary files or text files, since the

MATLAB *load* command accepts both types. However, loading operations are usually faster in binary format.

The formulas implemented for the different surfaces and the atmosphere are described below. Variables whose name starts by T correspond to brightness temperatures, while variables starting by R correspond to reflection coefficients. These variables are matrices of Npixels rows (Npixels = length of input vectors) and 2 columns. The first column corresponds to vertical polarization and the second column corresponds to horizontal polarization. In addition to comments in the functions, more detailed comments have been included in green for better reference and software maintenance.

B.1. Bare soil (lines117-142)

CODE:

```
h1  = 4.*k.^2.*SoilRoughness.^2 ;          % Effective Roughness Parameter.
                                           % h1=1 when SoilRoughness=0
epsS   =   3.1   +   17.36.*SoilMoisture   +   63.12.*SoilMoisture.^2   +   j.*(0.031   +
.65*SoilMoisture+ 20.42*SoilMoisture.^2);
                                           % Wang's model

RS = reflcoef(epsS, 0, pixelthetainc1);  % Fresnel reflection coefficient for bare soil
RB(:,1) = RS(:,1).*exp(-h1.*(cos(pixelthetainc1).^2));
                                           % Reflection Coefficient with roughness
RB(:,2) = RS(:,2).*exp(-h1.*(cos(pixelthetainc1).^2));
                                           % Reflection Coefficient with roughness
% Q factor models the cross-polarization due to surface roughness
Q = 0.35.*(1-exp(-0.6.*(SoilRoughness./100).^2.*f./1e9));
                                           % SoilRoughness (m  -> cm);
                                           % f (Hz -> GHz)
RS = [(1-Q).*RB(:,1)+Q.*RB(:,2) , (1-Q).*RB(:,2)+Q.*RB(:,1)];
```

B.2. Vegetation covered soils (lines 153-173)

The following parameters have been assumed:

- Dry mass fraction = 0.3;
- Salinity of water in vegetation = 10 psu;
- Leaves volume fraction = 1/80; and

- Leaves and vegetation dielectric constant by Matzler's and Kerr's formulas (file e_veget.m)

They can be modified or made pixel dependent by using appropriate input files.

CODE:

```
m_dry       = 0.3;                          % dry mass
m_fresh     = 1.0;

[eps_veg]   = e_veget(f, SoilSurfTemp(Modelo1)-273, m_dry, m_fresh);
[RAV]       = reflcoeg(1,real(eps_veg),pixelthetainc1);

KeV         = 4.34.*2.*k.*abs(imag(sqrt(eps_veg)));
n           = real(sqrt(eps_veg));

% transmission angle vegetation -soil
teta2       = asin(n.*sin(pixelthetainc1));
teta2       = teta2.*(imag(teta2)==0) + pi/2.*(imag(teta2)~=0);

% vegetation losses. L=1 when there is no vegetation canopy
L       = exp(KeV.*VegetalHeight.*sec(teta2));
T1      = zeros(N_PIXEL,2);
R1      = zeros(N_PIXEL,2);

T1(Modelo1,1)     =     (1-RAV(:,1))./(1-(RAV(:,1).*RS(:,1)./L.^2)).*((1+RS(:,1)./L).*(1-
1./L).*(1-VegetalAlbedus)+(1-RS(:,1))./L).*SoilSurfTemp(Modelo1);
T1(Modelo1,2)     =     (1-RAV(:,2))./(1-(RAV(:,2).*RS(:,2)./L.^2)).*((1+RS(:,2)./L).*(1-
1./L).*(1-VegetalAlbedus)+(1-RS(:,2))./L).*SoilSurfTemp(Modelo1);
% reflexion in the upper layer + air-vegetation transmission/attenuation
% vegetetation/reflexion soil/attenuation vegattion/transmission vegetation-air

R1(Modelo1,1) = (1-RAV(:,1)).^2 .* RS(:,1) ./ L.^2 + RAV(:,1) ;
R1(Modelo1,2) = (1-RAV(:,2)).^2 .* RS(:,2) ./ L.^2 + RAV(:,2) ;
```

B.3. Snow covered soils (lines 198-242)

The following parameters have been assumed:

- Soil temperature = -1°C;
- Dielectric constant for frosted soil = 3;
- Ice density = 0.916 g/cm^3; and

- Snow density = 0.24 g/cm^3.

They can be modified or made pixel dependent by using appropriate input files.

The snow extinction coefficient and albedo are computed at the function kekas.m. This function performs a linear interpolation of the plots found in pages 1606 and 1609 of "Microwave Remote Sensing", Ulaby et al.

CODE:

```
TempSoil    = 272;      % "Microwave Scattering Models..."  A. K. Fung p. 406
epsIS       = 3;        % epsilon iced soil
epsA        = 1;        % epsilon air

[KeN,a]     = kekas(f);% computes snow extinction and albedo.

albedo      = a.*SnowAlbedus; % the albedo is multiplied by the albedo data from the file

[Teffs,roeffs]=  stros(SnowDepth*100,TempSoil,f,KeN);% computes the effective surface
temperature and density, assuming an exponential temperature variation vs. depth within the
snow layer and the snow density vs. depth from TSANG et al (1994)

[epsN] = eps_ds(roeffs,f);          % Dry snow dielectric constant, from "Microwave
Scattering and Emission Models", A.K. Fung, 1994. Formula OK for f < 13 GHz. Ice density is
assumed to be 0.916 g/cm^3;

n       = sqrt(real(epsN));

theta2  = asin(n.*sin(pixelthetainc21));      % angle of transmission in snow layer

theta2  = theta2.*(imag(theta2)==0) + pi/2.*(imag(theta2)~=0);

L       = exp(KeN.*SnowDepth.*sec(theta2));   % snow attenuation

[RA_N]  = reflcoeg(epsA, real(epsN), pixelthetainc21);

h1      = 4.*k.^2.*SnowRoughness.^2 ;         % Effective Roughness Parameter.
                                              % h1=1 when SoilRoughness=0
RA_N1(:,1) = RA_N(:,1).*exp(-h1.*(cos(pixelthetainc21).^21));
                                              % Roughned soil reflection coefficient V-pol
RA_N1(:,2)     = RA_N(:,2).*exp(-h1.*(cos(pixelthetainc21).^21));
                                              % Roughned soil reflection coefficient H-pol

% Q factor models the cross-polarization due to surface roughness
```

```
    Q       = 0.35.*(1-exp(-0.6.*(SnowRoughness./100).^2.*f./1e9));
                                                % polarization mixing parameter
                                                % SnowRoughness (m  -> cm);
                                                % f (Hz -> GHz)

    RA_N    = [(1-Q).*RA_N1(:,1)+Q.*RA_N1(:,2) , (1-Q).*RA_N1(:,2)+Q.*RA_N1(:,1)];

    clear RA_N1

    [RN_S]  = reflcoeg(real(epsN), epsIS, theta2);

    T2s     = zeros(N_PIXEL,2);
    R2s     = zeros(N_PIXEL,2);

    % Radiative transfer equation for three layer system: air + snow + soil

    T2s(Modelo21,1)    = (1-RA_N(:,1))./
(1-RA_N(:,1).*RN_S(:,1)./L.^2)).*((1+RN_S(:,1)./L).*(1-1./L).*(1-SnowAlbedus).*Teffs
+ (1-RN_S(:,1))./L.*TempSoil);
    T2s(Modelo21,2)    = (1-RA_N(:,2))./
(1-(RA_N(:,2).*RN_S(:,2)./L.^2)).*((1+RN_S(:,2)./L).*(1-1./L).*(1-SnowAlbedus).*Teffs
+ (1-RN_S(:,2))./L.*TempSoil);

    R2s(Modelo21,1)    = (1-RA_N(:,1)).^2.* RN_S(:,1)./L.^2 + RA_N(:,1);
    R2s(Modelo21,2)    = (1-RA_N(:,2)).^2.* RN_S(:,2)./L.^2 + RA_N(:,2);
```

B.4. Frosted snow soil (lines 257-289)

The following parameters have been assumed:

- Soil temperature = -1°C;
- Dielectric constant for iced soil = 3; and
- Real part of pure ice dielectric constant = 3.15.

They can be modified or made pixel dependent by using appropriate input files.

The imaginary part of the dielectric constant of pure ice (no salt) is computed in the function eps_ice.m. This function performs two linear interpolations from 1 to 10 GHz and from 10 to 100 GHz of the plot in page 2027 of "Microwave Remote Sensing", Ulaby et al.

CODE:

```
TempIce = 272;          % "Microwave Scattering and Emission Models", A. K. Fung p. 406.
epsA    = 1;            % epsilon air
epsIS   = 3;            % epsilon frosted soil
epsIcei= 3.15;          % real part of pure ice dielectric constant

[epsIceii]=eps_ice(f);% imaginary part of pure ice dielectric constant

epsIce  = epsIcei+j.*epsIceii;

KeI     = 4.34.*2.*k.*abs(imag(sqrt(epsIce)));    % Attenuation [dB/m]

n       = sqrt(real(epsIce));
theta2  = asin(n.*sin(pixelthetainc22));          % trasnmission angle in snow layer
theta2  = theta2.*(imag(theta2)==0) + pi/2.*(imag(theta2)~=0);

L       = exp(KeI.*IceDepth.*sec(pixelthetainc22));% attenuation due frosted snow

[KeN,a] = kekas(f);         % snow extinction constant and albedo

albedo  = a.*IceAlbedus;    % the albedo is multiplied by the albedo data from the file
 [RA_I]     = reflcoeg(epsA,real(epsIce),pixelthetainc22);

[RI_S]  = reflcoeg(real(epsIce),epsIS,theta2);

T2i     = zeros (N_PIXEL,2);
L2i     = zeros (N_PIXEL,1);
R2i     = zeros (N_PIXEL,2);

% Radiative transfer equation for three layer system: air + iced snow + soil

T2i(Modelo22,1)    = ((1-RA_I(:,1))./
(1-(RA_I(:,1).*RI_S(:,1)./L.^2)).*((1+RI_S(:,1)./L).*(1-1./L).*(1-albedo)
+ (1-RI_S(:,1))./L).*TempIce);
T2i(Modelo22,2)    = ((1-RA_I(:,2))./
(1-(RA_I(:,2).*RI_S(:,2)./L.^2)).*((1+RI_S(:,2)./L).*(1-1./L).*(1-albedo)
+ (1-RI_S(:,2))./L).*TempIce);
R2i(Modelo22,1)    = (1-RA_I(:,1)).^2.* RI_S(:,1)./L.^2 + RA_I(:,1);
R2i(Modelo22,2)    = (1-RA_I(:,2)).^2.* RI_S(:,2)./L.^2 + RA_I(:,2);
```

B.5. Water bodies (oceans, seas and lakes) (lines 319-334)

The following assumptions have been made:

- The dielectric permitivity of salt water is computed using the Klein and Swift '77 model (functions epsilon.m and epsiloni.m). These functions can be modified with Ellison et al. '95 formulas, or more accurate expressions that can be developed at L-band in the future.
- The wind effects (roughness and foam) are modeled using a linear approximation of Hollinger's '71 data: $\Delta T_{\substack{h\\v}} = 0.2\left(1 \mp \frac{\theta}{55^\circ}\right) WS_{[m/s]}$. This functional dependence can be modified with more recent data (i.e. WISE 2000 and WISE 2001 campaigns).

- No polarimetric models (for the third and fourth Stoles elements) have been included.

CODE:

```
OceanSurfTempC= (OceanSurfTemp - 273.15);
OceanSurfTempC= OceanSurfTempC + (OceanSurfTempC==0).*eps;

% Klein and Swift dielectric model
[epsr]      = epsilon  (f, OceanSurfTempC, OceanSalinity);
[epsi]      = epsiloni (f, OceanSurfTempC, OceanSalinity);

[RO]        = reflcoef(epsr, epsi, pixelthetainc3 );

TO(:,1) = (1 - RO(:,1)).*OceanSurfTemp;
TO(:,2) = (1 - RO(:,2)).*OceanSurfTemp;

wind    = sqrt(ZonalWind.^2+MeridionalWind.^2);

TO(:,1) = TO(:,1)+ 0.2*(1-pixelthetainc3./(55*pi/180)).*wind;% Hollinger's linear fit
TO(:,2) = TO(:,2)+ 0.2*(1+pixelthetainc3./(55*pi/180)).*wind;
```

B.6. Iced oceans (lines 341-348)

The following assumption has been made:

- The dielectric permitivity of the ice is $\varepsilon_r = 3 - j\,0.25$. For first year ice ε_r ranges between $\varepsilon_r = 3 - j\,0.25$ and $\varepsilon_r = 3 - j\,0.07$, while for multiyear ice from $\varepsilon_r = 3 - j$

0.07 and ε_r = 3 – j 0.03 ("Microwave Remote Sensing", Ulaby et al., p 2058-2059).

This parameter can be changed if information of ice age is available.

Note that the spatial resolution of the ice cover fraction is 1°x1° while the coastlines and inner water bodies is 1°/12 x 1°/12. Therefore, the border between iced sea and liquid sea appears with this coarse resolution.

CODE:

```
EpsIOreal  = 3.0;
epsIOimag  = 0.25;

[RIO]      = reflcoef(epsIOreal ,epsIOimag, pixelthetainc3);

TIO(:,1)   = (1 - RIO(:,1)).*OceanSurfTemp;
TIO(:,2)   = (1 - RIO(:,2)).*OceanSurfTemp;
```

B.7. Atmosphere (lines 384-389)

The following assumptions have been made:

- Air pressure at sea level = 1013 mbar;
- Air temperature at 2 m = 10°C;
- Atmosphere height = 32 km (only important at very large incidence angles > 80° to account for Earth curvature effects).

They can be modified or made pixel dependent by using appropriate input files.

TU, TD and Latm stand for upwelling and downwelling atmospheric temperatures, and atmospheric attenuation. These approximate formulas have been obtained from linear regressions of Liebe's model at L-band (1413 MHz).

CODE:

```
Pressure   = 1013.*ones(size(Pressure));
```

```
    TETAEQ     =  acos(  hatm./(sqrt(RT.^2.*cos(pixelthetainc).^2  +(hatm+2.*RT).*hatm)  -
RT.*cos(pixelthetainc)));     % Equivalent incidence angle, taking into account Earth's
curvature effects
    TU(:,1)    = (1.86 + 0.0029.*RainRate + 0.10 .*W_Liquid + 0.0034.*(Pressure - 1013) -
0.011.* ((SurfTemp-283.15).*(SurfTemp-283.15>=273.15))./cos(TETAEQ);
    TD(:,1)    = (2.10 + 0.0029.*RainRate + 0.10 .*W_Liquid + 0.0034.*(Pressure - 1013) -
0.011.* ((SurfTemp-283.15).*(SurfTemp-283.15>=273.15))./cos(TETAEQ);
    Latm(:,1)  = (0.04024 + 0.000046.*RainRate + 0.0030.*W_Liquid + 0.000059.*(Pressure -
1013) - 0.00030.* (SurfTemp - 283.15))./cos(TETAEQ);
```

B.8. Ionosphere

The modeling of ionosphere effects requires the knowledge of the geomagnetic field and the total electron content. This is performed by functions igrf2000.m and iri95.m that call Fortran compiled MATLAB mex functions. These functions are described below. It is not possible to modify them.

```
function [babs,bnorth,beast,bdown,dip,dec,xl,icode]=igrf2000(xlat,xlong,year,height)

%function [babs,bnorth,beast,bdown,dip,dec,xl,icode]=igrf2000(xlat,xlong,year,height)
%
%MEX file computing the geomagnetic field, based in the International Geomagnetic
%Reference Field model. It is implemented in a set of Fortran functions, obtained
%from the National Space Science Data Center ftp server. These can be found at
%ftp://nssdc.gsfc.nasa.gov/pub/models/geomagnetic/igrf/fortran_code/
%
%  INPUT:
%    xlat   = geodatic latitude in degrees
%    xlong  = geodatic longitude in degrees
%    year   = decimal year (year+month/12.0-0.5 or year+day-of-year/365
%             or 366 if leap year)
%    height = height in km
%
%  OUTPUT:
%    babs   = magnetic field strength in tesla
%    bnorth, beast, bdown
%           = components of the field in tesla with respect to the local geodetic
%             coordinate system, with axis pointing in the tangential plane to the
%             north, east, and downward
%    dip    = geomagnetic inclination in degrees
%    dec    = geomagnetic declination in degress
%    xl     = L value
%    icode  = code indicating the validity of L value computation:
%                icode = 1 -->  L is correct;
```

```
%               icode = 2 -->  L is not correct;
%               icode = 3 -->  an approximation is used.

function         IRIA,IRIB]=iri95(jmag,jf,options,lat,long,year,mmdd,iut,dhour,height,htecmax,ivar,
varbegin,varend,varstep)
%function        IRIA,IRIB]=iri95(jmag,jf,options,lat,long,year,mmdd,iut,dhour,height,htecmax,ivar,
varbegin,varend,varstep)
%
%MEX file computing ionosphere parameters, based in the Intenational Reference Ionosphere '95
model.
%It has been implemented in a set of Fortran functions by Dieter Bilitza, and obtained from the
%National Space Science Data Center ftp server. The Fortran code can be found at
%ftp://nssdc.gsfc.nasa.gov/pub/models/ionospheric/iri/iri95/fortran_code/
%
%  INPUT:
%    jmag    = 0/1 indicating geographic/geomagnetic coordinates, respectively
%
%    jf      = set of 20 logical(0/1) values indicating model options:
%            jf(i)=          1                          0
%           ------------------------------------------------------------------
%            i=1         Ne computed            Ne not computed
%              2         Te, Ti computed        Te, Ti not computed
%              3         Ni computed            Ni not computed
%              4         B0 - Table option      B0 - Gulyaeva (1987)
%              5         foF2 - CCIR            foF2 - URSI
%              6         Ni - Standard          Ni - Danilov-Yaichnikov-Smironova
%              7         Ne - Standard Topside  Ne - IRI-79 Topside
%              8         foF2 from model        foF2 or NmF2 - user input
%              9         hmF2 from model        hmF2 or M3000F2 - user input
%             10         Te - Standard          Te - Using Te/Ne correlation
%             11         Ne - Standard Profile  Ne - Lay-function formalism
%             12         Unit for message displaying. It has been disabled for the MEX file
%                        implementation , in order not to cause MATLAB fails.
%             13         foF1 from model        foF1 or NmF1 - user input
%             14         hmF1 from model        hmF1 - user input
%             15         foE  from model        foE or NmE - user input
%             16         hmE  from model        hmE - user input
%             17         Rz12 from file         Rz12 - user input
%             18         simple ut<->lt         using ut_lt subroutine
%             19    free
%             20    free
%
%    options = optional input parameters, as specified by jf.
%
%                  Setting              INPUT parameter
%              ------------------------------------------------------------------
```

```
%                 jf(8) = 0      option(1)=user input for foF2 [MHz] or NmF2 [m-3]
%                 jf(9) = 0      option(2)=user input for hmF2 [km] or M(3000)F2
%                 jf(10)= 0      option(3),option(4),option(5)=user input for
%                                      Ne(300km),Ne(400km),Ne(600km) [m-3]
%                 jf(13)= 0      option(6)=user input for foF1 [MHz] or NmF1 [m-3]
%                 jf(14)= 0      option(7)=user input for hmF1 [km]
%                 jf(15)= 0      option(8)=user input for foE [MHz] or NmE [m-3]
%                 jf(16)= 0      option(9)=user input for hmE [km]
%                 jf(17)= 0      option(10)=user input for Rz12
%
%    lat     = latitude in coordinates chosen by jmag, in degrees. Range [-90º,+90º]
%    long    = longitude in coordinates chosen by jmag, in degrees. Range [0º,+360º]
%    year    = year of computed samples. Range [1958,2003]
%    mmdd    = month+day (MMDD. Range of MM [1,12]. Range of DD [1,days of month]),
%              or day of year (-DDD. Range [-1,-365/-366])
%    iut     = 0/1 indicating Local Time/Universal Time, respectively
%    dhour   = decimal hour in chosen day. Range [0,23.99]
%    height  = altitude of the observer, in km. Range [0, inf.*]
%    htecmax = maximum height for TEC integration, in km. range [0,inf.*]
%              If htecmax = 0 no TEC integration is computed.
%
%            *(Note: if height> 50000, a message is displayed to remind height units)
%
%                 THE ALTITUDE LIMITS ARE:  LOWER (DAY/NIGHT)  UPPER
%                      ELECTRON DENSITY         60/80 KM       1000 KM
%                      TEMPERATURES              120 KM        3000 KM
%                      ION DENSITIES             100 KM        1000 KM
%
%    ivar    = type of profile. Range [1,8]:
%              ivar = 1      altitude
%                   = 2,3    latitude,longitude
%                   = 4,5,6  year,month,day
%                   = 7      day of year
%                   = 8      hour (UT or LT)
%    varbegin= Lower boundar of selected profile
%    varend  = Upper boundary of selected profile
%    varstep = amplitude of interval for profile. Maximum number of steps: 100.
%
%  OUTPUT: not available values are set to -1
%
%    IRIA(1:11,1:100):
%
%         IRIA(1,1:100)   electron density [m-3]
%         IRIA(2,1:100)   neutral temperature [K]
%         IRIA(3,1:100)   ion temperature [K]
%         IRIA(4,1:100)   electron temperature [K]
%         IRIA(5,1:100)   O+ ion density [m-3]
```

```
%         IRIA(6,1:100)   H+ ion density [m-3]
%         IRIA(7,1:100)   HE+ ion density [m-3]
%         IRIA(8,1:100)   O2+ ion density [m-3]
%         IRIA(9,1:100)   NO+ ion density [m-3]
%              AND, IF JF(6) = 0 :
%         IRIA(10,1:100)  percentage of Cluster Ions in %
%         IRIA(11,1:100)  percentage of N+ Ions in %
%
%    IRIB(1:38,1:100):
%
%           Additional output parameters. Only the first NUMSTEP fields are eventually
%           filled for each parameter, where NUMSTEP =((varbegin-varend)/varstep)+1
%
%           IRIB(1,1:NUMSTEP)  = NmF2 [m-3]           IRIB(2,1:NUMSTEP)  = hmF2/KM
%           IRIB(3,1:NUMSTEP)  = NmF1 [m-3]           IRIB(4,1:NUMSTEP)  = hmF1/KM
%           IRIB(5,1:NUMSTEP)  = NmE [m-3]            IRIB(6,1:NUMSTEP)  = hmE/KM
%           IRIB(7,1:NUMSTEP)  = NmD [m-3]            IRIB(8,1:NUMSTEP)  = hmD/KM
%           IRIB(9,1:NUMSTEP)  = Hhalf [Km]           IRIB(10,1:NUMSTEP) = B0/KM
%           IRIB(11,1:NUMSTEP) = VALLEY-BASE [m-3]    IRIB(12,1:NUMSTEP) = valley-top [Km]
%           IRIB(13,1:NUMSTEP) = TE-peak [K]          IRIB(14,1:NUMSTEP) = TE-Peak height [Km]
%           IRIB(15,1:NUMSTEP) = TE-MOD(300KM)        IRIB(16,1:NUMSTEP) = TE-MOD(400KM) [K]
%           IRIB(17,1:NUMSTEP) = TE-MOD(600KM)        IRIB(18,1:NUMSTEP) = TE-MOD(1400KM) [K]
%           IRIB(19,1:NUMSTEP) = TE-MOD(3000KM)       IRIB(20,1:NUMSTEP) = TE(120KM)=TN=TI [K]
%           IRIB(21,1:NUMSTEP) = TI-MOD(430KM)        IRIB(22,1:NUMSTEP) = X [Km], where TE=TI
%           IRIB(23,1:NUMSTEP) = Sol Zenith Ang [º]   IRIB(24,1:NUMSTEP) = SUN Declination [º]
%           IRIB(25,1:NUMSTEP) = DIP [º]              IRIB(26,1:NUMSTEP) = DIP latitude [º]
%           IRIB(27,1:NUMSTEP) = Modified DIP lat.    IRIB(28,1:NUMSTEP) = DELA
%           IRIB(29,1:NUMSTEP) = sunrise [dec. hours] IRIB(30,1:NUMSTEP) = sunset [dec. hours]
%           IRIB(31,1:NUMSTEP) = Iseason (1=spring)   IRIB(32,1:NUMSTEP) = Nseason (northern)
%           IRIB(33,1:NUMSTEP) = Rz12                 IRIB(34,1:NUMSTEP) = Covington Index
%           IRIB(35,1:NUMSTEP) = B1                   IRIB(36,1:NUMSTEP) = M(3000)F2
%
%       place_holders only:
%           IRIB(37,1:NUMSTEP) = TEC/m-2            IRIB(38,1:NUMSTEP) = TEC_top/TEC*100.
%
%
%           **************              ALL ANGLES ARE IN DEGREE          **************
%           **************              ALL DENSITIES ARE IN M-3          **************
%           **************              ALL ALTITUDES ARE IN KM           **************
%           **************            ALL TEMPERATURES ARE IN KELVIN      **************
%           **************            ALL TIMES ARE IN DECIMAL HOURS      **************
```

APPENDIX C. USING SEPS TO ESTIMATE THE RMS RADIOMETRIC ACCURACY (PIXEL BIAS) SENSIBILITY TO INSTRUMENTAL ERRORS

With a lot of patience, SEPS can also be used to compute the impact of instrument's error in the radiometric accuracy. It is difficult to study isolatedly the impact of each error source, since SEPS includes many of them, but also an "imperfect" calibration and error correction algorithm (affected by thermal noise, and some limitations, such as the impossibility to calibrate non-separable errors for baselines not driven by the same noise source), and an image reconstruction algorithm.

The simplest way to analyze the sensibility to errors is the following:

1. Except for the parameter to be studied:
   - set all instrument errors to zero though the GUI (it includes use of different antenna position error files and different antenna pattern files for generation of observables and image reconstruction/auxiliary visibilities computation),
   - set all instrument errors to zero in .sbs parameters,
   - set Sun and Moon effects to "off",
   - set –eventually- "backlobe modelling" to off in Simulation Parameters Definition > Simulation Parameters,
   - set integration time equal to a very large number (correlations used in the calibration will be more accurate),
   - set integration time for NIR and PMS to a very large number (in NIR.sbs and PMS.sbs files),
   - set "Use inverse FFT for Image Reconstruction" to on in Simulation Parameters Definition > Timeline, and
   - activate the "instrument parameter random number generator" and "different state" boxes in the "Mission Simulation" menu, and
   - activate the "Computation of Instrument's Performance (ESA SRD Definition)."

2. Run at least 10 Monte Carlo simulation for each parameter value
3. Load the radiometric accuracy (or bias) at a given polarization from file SMOS_PERFORMANCE_SRD_DEF.txt

from the path:

ROOT:\SEPSv4\Data\Output\Bin\SCENARIO_NAME for the SCENARIO_NAME being simulated.

4. Compute the mean value of the radiometric accuracy (or bias).
5. Repeat steps 1-4 for increasing parameter values.
6. Plot the mean values computed vs. the parameter value (error source):
   - At low parameter values (small errors) a saturation effect takes place, since some effects are still present (discretization, imperfect calibration etc.)
   - At medium parameter values the trend becomes linear, and the slope is the "sensibility" to that given parameter we are looking for.
   - Finally, at high parameter values non-linearities start appearing and the linear trend (parameter value – mean radiometric accuracy) is no longer followed.

   The location of these transition knees depends on the particular type of error.

The process described above is very tedious, but can be heavily alleviated by using the "SEPS Batch mode".

APPENDIX D. FLAT TARGET TRANSFORMATION: SIMULATING MULTIPLE SKY POINTING MEASUREMENTS

The flat target transformation is a technique to compensate most of the errors induced by an imperfect antenna pattern characterization and relies on deep sky measurements, which are referred as "Flat Target". It was originally devised by M. Martín-Neira [Martín-Neira, 2004] to operate in the alias-free FOV, and it was later extended by M. Suess [M. Suess, 2005] to perform in the extended alias-free FOV. The interested reader is referred to these publications for details on the principles and the implementation of the method.

The possibility of computing the "flat target response" from multiple sky observations, in which the galactic background, the Sun and the Moon can be varying was not requested during the SEPS development, and it only appeared during the software review process. This appendix shows how easily the user can adapt the current SEPS for this purpose. The steps are described below.

1) Creating a simulation consisting ONLY in as many FTT events as needed (see timeline defintion, p. 109).
2) Adding a counter 'NNN' in the file (lines 143 to 149)

    ..\Exec\S_functions\Observables\computeFTTcal.m

    so that files visibvarsFTT.mat and ref_imags_FTT.mat are name instead:

    visibvarsFTT_NNN.mat and ref_imags_FTT_NNN.mat

3) In Matlab, but outside SEPS, load the contents of the variables in file visibvarsFTT_NNN.mat with the load command and average all them.
4) Store the variables in a file called visibvarsFTT.mat, and copy this file to the output directory of an existing scenario, for example:
..\Data\Output\Bin\BD_01_Nominal_Attitude_Indian_Ocean_G_12_sec
5) Repeat steps #3 and #4 for the variables in file ref_imags_FTT_NNN.mat, storing them in file ref_imags_FTT.mat.
4) Finally, run a new scenario selecting in the option "Initial calibration files inherited from a previously executed scenario" in the **SIMULATION PARAMETERS DEFINITION** menu, under the ***Timeline*** tab, the FTT files of the output directory in which you have previously stored the averaged values (steps #4 and #5).

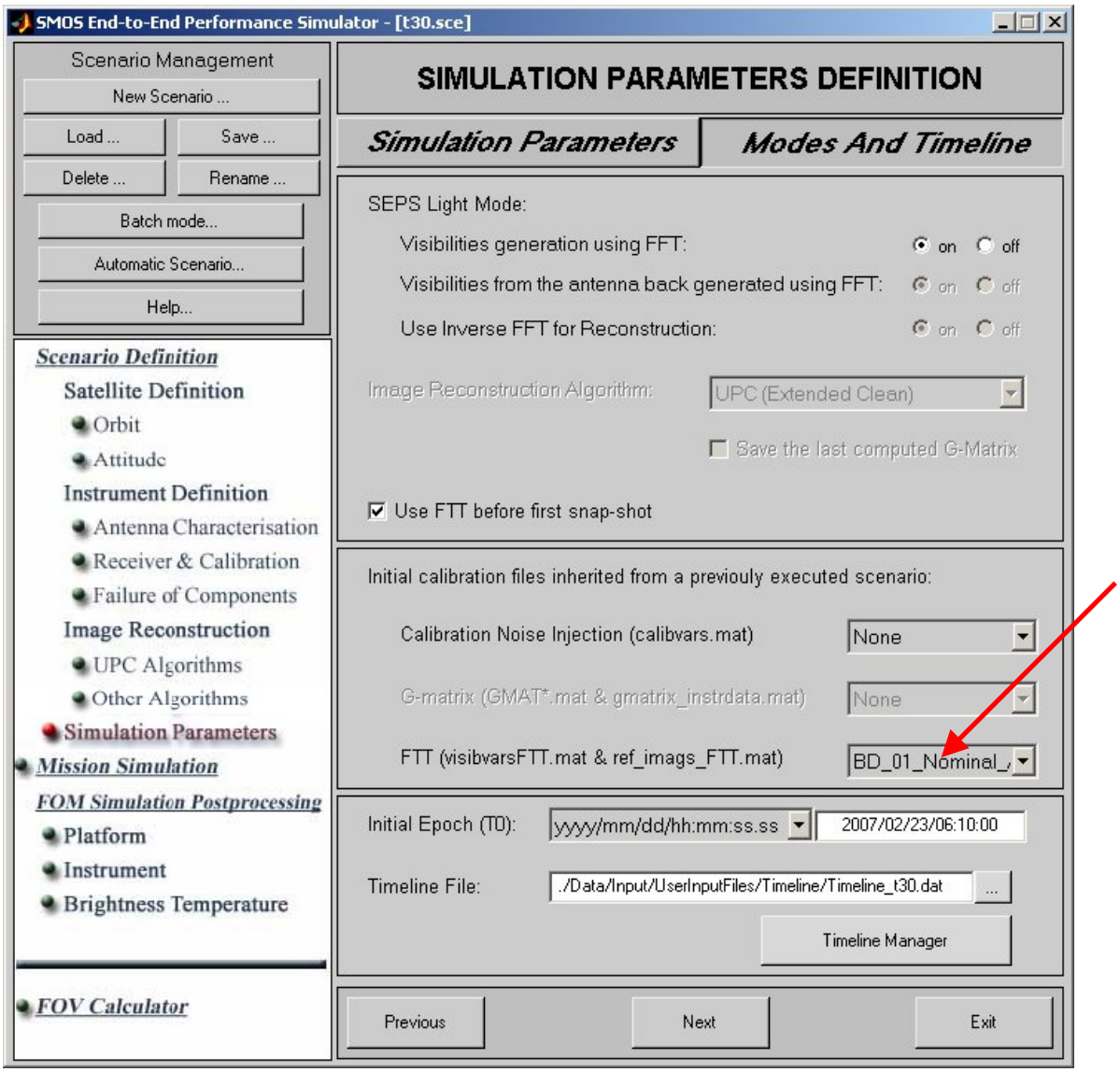


Figure D.1. Selecting initial calibration files from previously executed scenarios. This option can be used to simulate the FTT when many sky measurements are averaged to obtain the Flat Target Response.

REFERENCES

J. Bará, A. Camps, F. Torres, and I. Corbella [1998], "Angular Resolution of Two-Dimensional Hexagonally Sampled Interferometric Radiometers," *Radio Science,* **33**, 5, pp. 1459-1473.

A. Camps [1996], "Application of Interferometric Radiometry to Earth Observation," Ph. D. Dissertation, Universitat Politècnica de Catalunya [http://www.edicionsupc.es/forum /fautcat.htm], pp. 17-30.

A. Camps, J. Bará, I. Corbella, and F. Torres [1997a], "The Processing of Hexagonally Sampled Signals with Standard Rectangular Techniques: Application to 2D Large Aperture Synthesis Interferometric Radiometers," *IEEE Transactions on Geoscience and Remote Sensing*, **GRS-35**, 1, pp. 183-190.

A. Camps, F. Torres, J. Bará, I. Corbella, and F. Monzón [1999], "Automatic Calibration of Channels Frequency Response in Interferometric Radiometers," *Electronics Letters*, **35**, 2, pp. 115-116.

J. P. Claassen and A. K. Fung [1974], "The Recovery of Polarized Apparent Temperature Distributions of Flat Scenes from Antenna Temperature Measurements," *IEEE Transactions on Antennas and Propagation*, **AP-22**, 3, pp. 433-442.

J. W. Goodman [1968], *Introduction to Fourier Optics*, ed. Mc Graw-Hill.

J. W. Goodman [1985], *Statistical Optics*, ed. Wiley Interscience.

A. C. Ludwig [1973], "The Definition of Cross-Polarization," *IEEE Transactions on Antennas and Propagation*, **AP-21**, 1, pp. 116-119.

M. Martín-Neira and J.M. Goutoule [1997], "A Two-Dimensional Aperture-Synthesis Radiometer for Soil Moisture and Ocean Salinity Observations," *ESA Bulletin*, no. 92, pp. 95-104.

M. Martín-Neira and Q. García [1999], "POL-SWITCHING: Switching Scheme for Single Channel Polarimetric Aperture Synthesis Radiometers," ESTEC Working Paper, No 2062.

M. Martín-Neira [2001], "Introduction to Two-Dimensional Aperture Synthesis Microwave Radiometry for Earth Observation: Polarimetric Formulation of the Visibility Function," ESTEC Working Paper, No 2130, ESA.

M. Martin-Neira [2004] "In-Orbit External Calibration and Validation" TEC-ETP/2004.103/MMN, Date 27-Oct-2004

R. M. Mersereau [1979], "The Processing of Hexagonally Sampled Signals," *Proceedings of the IEEE*, **67**, 6, pp. 930-949.

W. H. Peake, W.H. [1959], "Interaction of Electromagnetic Waves with Some Natural Surfaces," *IRE Transactions on Antennas and Propagation*, **AP-7**, pp. S324-S329.

K. Rautiainen, R. Butora, T. Auer, I. Mononen, J. Salminen, S. Tauriainen, M. Hallikainen, J. Uusitalo, and P. Jukkala [2000], "The Helsinki University of Technology / Ylinen Electronics Airborne L-band Interferometric Radiometer," *Proceedings of the IEEE IGARSS'00*, Honolulu, (available from IEEE, 445 Hoes Lane, Piscataway, NJ 08855-1331, USA), **7**, pp. 2978-2980.

P. Silvestrin, M. Berger, Y. Kerr, and J. Font [2001], "ESA's Second Earth Explorer Opportunity Mission: The Soil Moisture and Ocean Salinity Mission–SMOS," *IEEE Geoscience and Remote Sensing Newsletter*, Cumulative Issue 118, March 2001, pp. 11-14.

N. Skou [1989], "Microwave Radiometer Systems: Design and Analysis," Artech House, pp. 81-96.

M. Suess [2005], Reconstruction of the Alias Regions with the FTT, TOS-EDP/2005.001.v1/MS, Date 5-February-2005

A. R. Thompson, J. M. Moran, and G. W. Swenson [1986], *Interferometry and Synthesis in Radio Astronomy*," John Wiley and Sons, pp. 41-77.

F. T. Ulaby, R. K. Moore, and A. K. Fung [1981; 1982; 1986], *Microwave Remote Sensing, Active and Passive. Vols. I, II, III,* Norwood, MA, Artech House, **1**, pp. 186-255, 344-430, **2**, pp. 816-920, **3**, pp. 1522-1634.

S. H. Yueh, R. Kwok, F. K. Li, S. V. Nghiem, W. J. Wilson, and J. A. Kong [1994], "Polarimetric Passive Remote Sensing of Ocean Wind Vectors," *Radio Science*, **29**, 4, pp. 799-814.

S. H. Yueh, W. J. Wilson, F. K. Li, S. V. Nghiem, and W. B. Ricketts [1997], "Polarimetric Brightness Temperatures of Sea Surfaces Measured with Aircraft K- and Ka-Band Radiometers," *IEEE Transactions on Geoscience and Remote Sensing*, **GRS-35**, 5, pp. 1177-1187.

SO-TN-CASA-PLM-0279 "SMOS PLM Command and Control Document"

SO-TN-CASA-PLM-0549 "Ancillary Packet format"

SO-TN-CASA-PLM-0854 v1.1 "NIR model implementation in SEPS"